

SEGURIDAD DE DISPOSITIVOS MÓVILES: ANDROID 7.x



Septiembre de 2018

Edita:



CENTRO CRIPTOLOGICO NACIONAL
2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC, ou=sello
electrónico, serialNumber=S2800155J, o=CENTRO
CRYPTOLOGICO NACIONAL, cn=CENTRO CRIPTOLOGICO
NACIONAL, c=ES
2018.09.27 17:53:52 +02'00'

© Centro Criptológico Nacional, 2018

NIPO: 083-19-015-2

Fecha de Edición: septiembre de 2018

Mónica Salas y Raúl Siles (DinoSec) han participado en la realización y modificación del presente documento y sus anexos.

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

Quedan rigurosamente prohibidas, sin la autorización escrita del Centro Criptológico Nacional, bajo las sanciones establecidas en las leyes, la reproducción parcial o total de este documento por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares del mismo mediante alquiler o préstamo públicos.

PRÓLOGO

El uso masivo de las tecnologías de la información y las telecomunicaciones (TIC), en todos los ámbitos de la sociedad, ha creado un nuevo espacio, el ciberespacio, donde se producirán conflictos y agresiones, y donde existen ciberamenazas que atentarán contra la seguridad nacional, el estado de derecho, la prosperidad económica, el estado de bienestar y el normal funcionamiento de la sociedad y de las administraciones públicas.

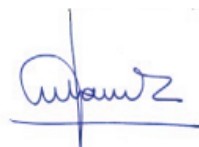
La Ley 11/2002, de 6 de mayo, reguladora del Centro Nacional de Inteligencia (CNI), encomienda al Centro Nacional de Inteligencia el ejercicio de las funciones relativas a la seguridad de las tecnologías de la información en su artículo 4.e), y de protección de la información clasificada en su artículo 4.f), a la vez que confiere a su Secretario de Estado Director la responsabilidad de dirigir el Centro Criptológico Nacional (CCN) en su artículo 9.2.f).

Partiendo del conocimiento y la experiencia del CNI sobre amenazas y vulnerabilidades en materia de riesgos emergentes, el Centro realiza, a través de su Centro Criptológico Nacional, regulado por el Real Decreto 421/2004, de 12 de marzo, diversas actividades directamente relacionadas con la seguridad de las TIC, orientadas a la formación de personal experto, a la aplicación de políticas y procedimientos de seguridad, y al empleo de tecnologías de seguridad adecuadas.

El Real Decreto 3/2010, de 8 de enero, por el que se regula el Esquema Nacional de Seguridad en el ámbito de la Administración Electrónica (ENS, en adelante), al que se refiere el apartado segundo del artículo 156 de la Ley 40/2015, de 1 de octubre, de Régimen Jurídico del Sector Público, establece la política de seguridad en la utilización de medios electrónicos que permita una protección adecuada de la información.

Precisamente el Real Decreto 3/2010 de 8 de Enero, modificado por el Real Decreto 951/2015, de 23 de octubre, fija los principios básicos y requisitos mínimos así como las medidas de protección a implantar en los sistemas de la Administración, y promueve la elaboración y difusión de guías de seguridad de las tecnologías de la información y las comunicaciones (STIC) por parte de CCN para facilitar un mejor cumplimiento de dichos requisitos mínimos.

En definitiva, la serie de documentos CCN-STIC se elabora para dar cumplimiento a los cometidos del Centro Criptológico Nacional y a lo reflejado en el Esquema Nacional de Seguridad, conscientes de la importancia que tiene el establecimiento de un marco de referencia en esta materia que sirva de apoyo para que el personal de la Administración lleve a cabo su difícil, y en ocasiones, ingrata tarea de proporcionar seguridad a los sistemas de las TIC bajo su responsabilidad.



Septiembre de 2018

Félix Sanz Roldán
Secretario de Estado
Director del Centro Criptológico Nacional

ÍNDICE

1. INTRODUCCIÓN	7
2. OBJETO	7
3. ALCANCE	9
4. HISTORIA DE ANDROID: VERSIONES Y CARACTERÍSTICAS.....	9
4.1. ANDROID 2.X.....	11
4.2. ANDROID 3.X.....	11
4.3. ANDROID 4.0.....	12
4.4. ANDROID 4.1.....	12
4.5. ANDROID 4.2.....	13
4.6. ANDROID 4.3.....	14
4.7. ANDROID 4.4.....	16
4.8. ANDROID 5.X.....	18
4.9. ANDROID 6.X.....	20
4.10. ANDROID 7.X.....	25
4.11. DIFERENCIAS ENTRE LAS MÚLTIPLES VERSIONES DE ANDROID.....	32
5. ENTORNO DE APLICACIÓN DE ESTA GUÍA.....	34
6. ACTUALIZACIÓN DEL SISTEMA OPERATIVO ANDROID.....	37
6.1. ACTUALIZACIÓN AUTOMÁTICA DEL SISTEMA OPERATIVO ANDROID.....	38
6.2. SERVICIOS DE GOOGLE PLAY	52
6.3. ACTUALIZACIÓN MANUAL COMPLETA DE ANDROID	52
6.4. ACTUALIZACIÓN MANUAL PARCIAL DE ANDROID.....	59
6.5. A/B (SEAMLESS) SYSTEM UPDATES	64
6.5.1. PROCESO DE ACTUALIZACIÓN SEAMLESS UPDATE	69
7. MODELO Y ARQUITECTURA DE SEGURIDAD DE ANDROID.....	70
7.1. MODELO DE PERMISOS DE ANDROID	71
7.1.1. MODELO DE PERMISOS EN TIEMPO DE INSTALACIÓN.....	72
7.1.2. MODELO DE PERMISOS EN TIEMPO DE EJECUCIÓN	76
7.1.3. PERMISOS NORMALES Y PELIGROSOS.....	80
7.1.4. PERMISOS SIMPLIFICADOS Y GRUPOS DE PERMISOS DE GOOGLE PLAY	80
7.1.5. PERMISOS DE ACCESO ESPECIAL	84
7.2. APP OPS	85
7.3. SANDBOX DE EJECUCIÓN DE LAS APPS	86
7.4. COMPARTICIÓN DE DATOS ENTRE APPS Y DIRECT SHARE	89
7.5. ART (ANDROID RUNTIME)	91
7.6. INSTALACIÓN DE APPS (DE ORÍGENES DESCONOCIDOS) EN ANDROID	93
7.7. FIRMA DIGITAL DE APPS EN ANDROID	95
7.7.1. ESQUEMA DE FIRMADO DE APK V1 (JAR).....	96
7.7.2. ESQUEMA DE FIRMADO DE APK V2.....	97
7.8. SELINUX EN ANDROID.....	98
7.9. PROCESO DE ARRANQUE SEGURO: VERIFIED BOOT	101
7.10. MODO DE ARRANQUE SEGURO: SAFE MODE	104
7.11. ANDROID WEBVIEWS	106
7.12. CONFIGURACIÓN POR DEFECTO DEL DISPOSITIVO MÓVIL.....	109

8. GESTIÓN EMPRESARIAL DE DISPOSITIVOS MÓVILES BASADOS EN ANDROID	111
8.1. GESTIÓN REMOTA DE DISPOSITIVOS ANDROID POR PARTE DEL USUARIO	114
8.2. ANDROID FOR WORK.....	131
9. ACCESO FÍSICO AL DISPOSITIVO MÓVIL - INTERFAZ DE USUARIO	136
9.1. PIN DE LA TARJETA SIM	136
9.2. CÓDIGO DE ACCESO AL DISPOSITIVO MÓVIL	139
9.3. HUELLA DACTILAR DIGITAL (<i>FINGERPRINT</i>).....	149
9.4. BLOQUEO Y DESBLOQUEO DE PANTALLA	155
9.5. DESBLOQUEO MEDIANTE SMART LOCK	158
9.6. PANTALLA DE DESBLOQUEO: PREVISUALIZACIÓN DE DATOS.....	170
9.7. INTERRUPCIONES.....	173
9.8. NOTIFICACIONES.....	177
9.9. AHORRO DE BATERÍA (DOZE)	182
9.10. CONFIGURADOR DE IU DEL SISTEMA	186
9.11. PANTALLA DE DESBLOQUEO: APAGAR EL DISPOSITIVO MÓVIL.....	189
9.12. MENÚ DE AJUSTES RÁPIDOS Y BARRA DE NOTIFICACIONES.....	191
9.13. PANTALLA DE DESBLOQUEO: WIDGETS Y CARPETAS.....	199
9.14. PANTALLA DE DESBLOQUEO: CÁMARA.....	202
9.15. PANTALLA DE DESBLOQUEO: INFORMACIÓN DEL PROPIETARIO	203
9.16. PRECAUCIONES Y RECOMENDACIONES ADICIONALES	205
9.17. DESBLOQUEO MEDIANTE LA CUENTA DE USUARIO DE GOOGLE	206
9.18. RESTAURACIÓN DEL CÓDIGO DE ACCESO	206
9.19. MÚLTIPLES PERFILES DE USUARIO	207
9.20. FACTORY RESET PROTECTION (FRP)	216
9.21. TALKBACK.....	217
10. CIFRADO DE DATOS EN ANDROID.....	217
10.1. CIFRADO DEL DISPOSITIVO MÓVIL	217
10.2. ALMACENAMIENTO ADOPTABLE.....	223
10.2.1. ALMACENAMIENTO ADOPTABLE EXTERNO	224
10.2.2. ALMACENAMIENTO ADOPTABLE INTERNO.....	226
10.3. DISPOSITIVOS DE ALMACENAMIENTO OTG (ON-THE-GO).....	232
10.4. ELIMINACIÓN DE DATOS DEL DISPOSITIVO MÓVIL.....	234
10.5. DIRECT BOOT	237
11. GESTIÓN DE CERTIFICADOS DIGITALES Y CREDENCIALES	239
11.1. ANDROID KEYSTORE SYSTEM: ALMACENAMIENTO DE CREDENCIALES.....	239
11.2. CERTIFICADOS DIGITALES DE SISTEMA Y DE USUARIO	243
11.3. GESTIÓN DE CERTIFICADOS DIGITALES	250
11.3.1. GESTIÓN DE CERTIFICADOS DIGITALES DE AUTORIDADES CERTIFICADORAS.....	250
11.3.2. GESTIÓN DE CERTIFICADOS DIGITALES SERVIDOR	255
11.3.3. GESTIÓN DE CERTIFICADOS DIGITALES CLIENTE	256
11.4. CERTIFICATE PINNING Y CERTIFICATE BLACKLISTING.....	259
12. LOCALIZACIÓN (O UBICACIÓN) GEOGRÁFICA.....	260
12.1. SERVICIOS DE UBICACIÓN DE GOOGLE	265
12.2. GOOGLE MAPS.....	267
12.3. A-GPS	275
12.4. GPSONEXTRA	276
12.5. HISTORIAL DE UBICACIONES.....	276

12.6. OTRAS APPS QUE HACEN USO DE LA LOCALIZACIÓN.....	281
12.6.1. BUSCADOR DE GOOGLE.....	281
12.6.2. REDES SOCIALES.....	283
12.6.3. NAVEGADOR WEB	283
12.7. INFORMACIÓN GEOGRÁFICA EN FOTOGRAFÍAS	285
13. COMUNICACIONES USB	288
13.1. ACCESO AL ALMACENAMIENTO POR USB.....	289
13.2. CONEXIÓN DE RED COMPARTIDA POR USB (<i>TETHERING</i>)	292
13.3. OPCIONES DE DESARROLLO Y DEPURACIÓN USB	295
13.4. RELACIONES DE CONFIANZA A TRAVÉS DE USB	298
14. COMUNICACIONES NFC.....	300
14.1. UTILIZACIÓN DE LAS COMUNICACIONES NFC	300
14.2. ANDROID BEAM.....	302
14.3. PAGOS (Y OTROS SERVICIOS) MEDIANTE NFC.....	305
15. COMUNICACIONES BLUETOOTH.....	310
15.1. UTILIZACIÓN DE LAS COMUNICACIONES BLUETOOTH.....	310
15.2. CONFIGURACIÓN GENERAL DE BLUETOOTH.....	314
15.3. DISPOSITIVOS BLUETOOTH EMPAREJADOS	316
15.4. CONFIGURACIÓN AVANZADA DE BLUETOOTH	321
15.5. OBEX PUSH.....	322
15.6. MANOS LIBRES BLUETOOTH.....	325
15.7. CONEXIÓN DE RED COMPARTIDA POR BLUETOOTH.....	326
15.8. BLUETOOTH LOW ENERGY (BLE)	328
15.9. GOOGLE NEARBY	331
16. COMUNICACIONES WI-FI	332
16.1. UTILIZACIÓN Y CONFIGURACIÓN DE LAS COMUNICACIONES WI-FI	332
16.2. SELECCIÓN DE LA RED DE DATOS EN ANDROID	339
16.3. RECOMENDACIONES DE SEGURIDAD PARA LA CONEXIÓN A REDES WI-FI.....	343
16.4. LISTA DE REDES PREFERIDAS (PNL).....	350
16.5. CONEXIÓN AUTOMÁTICA A REDES WI-FI: OCULTAS O VISIBLES Y RESTRICCIÓN DE CONEXIÓN	353
16.6. PUNTO DE ACCESO O ZONA WI-FI.....	358
16.7. WI-FI DIRECT ("P2P")	362
17. COMUNICACIONES MÓVILES: MENSAJES DE TEXTO (SMS).....	365
18. COMUNICACIONES MÓVILES: VOZ Y DATOS	372
18.1. DESACTIVACIÓN DE LAS COMUNICACIONES DE VOZ Y DATOS MÓVILES	372
18.1.1. DATA SAVER.....	377
18.2. CONFIGURACIÓN Y SELECCIÓN DE LA RED (VOZ) DE TELEFONÍA MÓVIL.....	380
18.3. CONFIGURACIÓN DE LA RED DE DATOS MÓVILES	388
18.4. SELECCIÓN DE LA RED DE DATOS MÓVILES	390
19. COMUNICACIONES TCP/IP	392
19.1. ADAPTADORES DE RED	392
19.2. SEGURIDAD EN TCP/IP	396
19.3. ESCANEADO DE PUERTOS TCP Y UDP.....	397
19.4. HTTPS: SSL/TLS.....	398
19.5. IPV6.....	402

19.6. REDES VPN	403
19.7. CONEXIÓN GTALKSERVICE	410
20. GESTIÓN DE APPS EN ANDROID	411
20.1. ACTUALIZACIÓN DE APPS EN ANDROID	411
20.2. ACTUALIZACIÓN AUTOMÁTICA DE APPS.....	419
20.3. INSTALACIÓN REMOTA DE APPS	421
20.4. ELIMINACIÓN REMOTA DE APPS	423
20.5. ESTADÍSTICAS DETALLADAS DE USO DE LAS APPS	423
20.6. ACCESO Y ALMACENAMIENTO DE DATOS DESDE APPS	425
20.7. ALMACENAMIENTO DE CREDENCIALES E INFORMACIÓN SENSIBLE EN APPS...	426
20.8. SERVICIO DE CONTRASEÑAS DE GOOGLE SMART LOCK	427
20.9. ENLACES Y REFERENCIAS WEB DESDE LAS APPS	429
20.10. INSTANT APPS.....	432
21. COPIAS DE SEGURIDAD	434
21.1. COPIA DE SEGURIDAD LOCAL	434
21.2. COPIA DE SEGURIDAD EN LOS SERVIDORES DE GOOGLE.....	436
22. INTERFAZ DE USUARIO DE ANDROID 7.X (NOUGAT)	439
22.1. PANTALLA DE AJUSTES	439
22.2. SPLIT-SCREEN.....	440
22.3. SCREEN PINNING	442
22.4. CAPTURA DE LA PANTALLA EN ANDROID.....	444
23. APÉNDICE A: PROCESO DE INSTALACIÓN Y CONFIGURACIÓN INICIAL.....	446
23.1. INSTALACIÓN SIN VINCULACIÓN CON CUENTA DE GOOGLE	451
23.2. INSTALACIÓN VINCULADA A UNA CUENTA DE GOOGLE.....	457
23.2.1. CREACIÓN DE UNA CUENTA DE USUARIO NUEVA	457
23.2.2. UTILIZACIÓN DE CUENTA DE GOOGLE EXISTENTE	460
24. REFERENCIAS	462

1. INTRODUCCIÓN

1. El desarrollo de los dispositivos y comunicaciones móviles, y de las tecnologías inalámbricas, en los últimos años ha revolucionado la forma de trabajar y comunicarse. El uso creciente de estas tecnologías sitúa a los dispositivos móviles como uno de los objetivos principales de las ciberamenazas.
2. La proliferación de dispositivos móviles en los últimos años, junto al aumento de las capacidades, prestaciones y posibilidades de utilización de los mismos, hace necesario evaluar en profundidad la seguridad ofrecida por este tipo de dispositivos respecto a la información que gestionan, profundizando en los mecanismos de protección que ofrecen, dentro de los entornos de Tecnologías de la Información y las Comunicaciones (TIC).
3. Se considera dispositivo móvil aquel dispositivo electrónico de uso personal o profesional de reducido tamaño que permite la gestión de información (almacenamiento, intercambio y procesamiento de información) y el acceso a redes de comunicaciones y servicios remotos, tanto de voz como de datos, y que habitualmente dispone de capacidades de telefonía, como por ejemplo teléfonos móviles, *smartphones* (teléfonos móviles avanzados o inteligentes), *tablets* (o tabletas) y agendas electrónicas (PDAs, Personal Digital Assistants), independientemente de si disponen de teclado físico o pantalla táctil.
4. Pese a que los dispositivos móviles se utilizan para establecer comunicaciones personales y profesionales, privadas y relevantes, y para el almacenamiento e intercambio de información sensible, el nivel de percepción de la amenaza de seguridad real existente no ha tenido la suficiente trascendencia en los usuarios finales y las organizaciones.
5. La concienciación, el sentido común y las buenas prácticas en la configuración y el uso de los dispositivos móviles constituyen una de las mejores defensas para prevenir y detectar este tipo de incidentes y amenazas [Ref.- 404].
6. El presente documento realiza un análisis detallado de los mecanismos y la configuración de seguridad recomendados para uno de los principales sistemas operativos de dispositivos móviles utilizados en la actualidad, Android 7.x, en concreto hasta la versión 7.1.2, con el objetivo de reducir su superficie de exposición frente a ataques de seguridad.

2. OBJETO

7. El propósito del presente documento es definir una lista de recomendaciones de seguridad para la configuración de dispositivos móviles basados en el sistema operativo Android versión 7.1.2 (en adelante Android) [Ref.- 1], cuyo objetivo es proteger el propio dispositivo móvil, sus comunicaciones y la información y datos que gestiona y almacena.
8. La presente guía proporciona los detalles específicos de aplicación e implementación de las recomendaciones de seguridad generales descritas

en la guía inicial de esta misma serie [Ref.- 400], que presenta la información necesaria para la evaluación y análisis de los riesgos, amenazas, y vulnerabilidades de seguridad a las que están expuestos los dispositivos móviles en la actualidad.

9. La serie CCN-STIC-450, "Seguridad de dispositivos móviles", se ha estructurado en dos niveles: una guía genérica inicial centrada en el análisis de seguridad de dispositivos móviles (CCN-STIC-450) [Ref.- 400], complementada por guías específicas para los principales sistemas operativos empleados por los dispositivos móviles hoy en día (o en el pasado). Por este motivo, antes de la lectura y aplicación de la presente guía, asociada a un sistema operativo (y versión concreta del mismo), se recomienda la lectura de la guía general de seguridad, CCN-STIC-450.
10. La presente guía está basada en la versión previa de la misma guía "Seguridad de dispositivos móviles: Android 6.x" [Ref.- 406], centrada en la versión 6.0.1 de Android y publicada en el año 2018.
11. Adicionalmente, se recomienda la lectura de las guías de esta misma serie asociadas a otros sistemas operativos y versiones de plataformas móviles, en caso de ser necesaria su aplicación en otros terminales, y a la gestión empresarial de dispositivos móviles (MDM):
 - CCN-STIC-450 - Seguridad de dispositivos móviles [Ref.- 400]
 - CCN-STIC-451 - Seguridad de dispositivos móviles: Windows Mobile 6.1
 - CCN-STIC-452 - Seguridad de dispositivos móviles: Windows Mobile 6.5
 - CCN-STIC-453(A) - Seguridad de dispositivos móviles: Android 2.x [Ref.- 401]
 - CCN-STIC-453B - Seguridad de dispositivos móviles: Android 4.x [Ref.- 402]
 - CCN-STIC-453C - Seguridad de dispositivos móviles: Android 5.x [Ref.- 405]
 - CCN-STIC-453D - Seguridad de dispositivos móviles: Android 6.x [Ref.- 406]
 - CCN-STIC-454(A) - Seguridad de dispositivos móviles: iPad (iOS 7.x)
 - CCN-STIC-455(A) - Seguridad de dispositivos móviles: iPhone (iOS 7.x)
 - CCN-STIC-457 - Gestión de dispositivos móviles: MDM (*Mobile Device Management*) [Ref.- 403]
 - CCN-STIC-456 - Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android [Ref.- 407]

Nota: Esta serie de guías están diseñadas considerando como requisito la necesidad de encontrar un equilibrio entre seguridad y usabilidad en relación a las capacidades y funcionalidad disponibles en los dispositivos móviles a proteger, con el objetivo de poder hacer uso de la mayoría de características disponibles en los mismos de forma segura.

3. ALCANCE

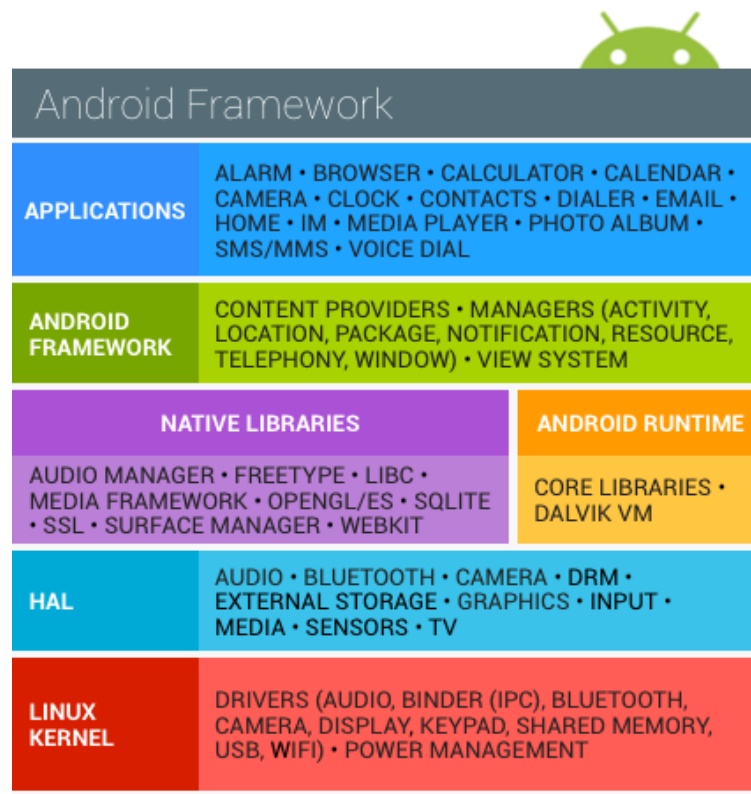
12. Las Autoridades responsables de la aplicación de la Política de Seguridad de las TIC (STIC) determinarán su análisis y aplicación a los dispositivos móviles ya existentes o futuros bajo su responsabilidad.

4. HISTORIA DE ANDROID: VERSIONES Y CARACTERÍSTICAS

13. Android es un sistema operativo diseñado para dispositivos móviles, que emplea software de código abierto (*open-source*) basado en Linux (originalmente en el kernel 2.6.x) y que incluye el sistema operativo y el kernel, una capa de abstracción hardware (HAL, *Hardware Abstraction Layer*), un conjunto de librerías (o APIs, *Application Programming Interface*), el entorno de ejecución (Dalvik y ART, *Android RunTime*) y el entorno de desarrollo para las aplicaciones móviles (de aquí en adelante referenciadas como apps, indistintamente, y por abreviar), tanto las desarrolladas por terceros como el conjunto de aplicaciones móviles (o apps) disponibles por defecto [Ref.- 11] (ver siguiente imagen¹).
14. Inicialmente desarrollado por Android Inc., fue adquirido por Google en el año 2005, colaborando en su desarrollo la Open Handset Alliance (OHA), y encargándose de su mantenimiento el Android Open Source Project (AOSP) [Ref.- 84]. La primera versión comercial, Android 1.0, fue publicada en septiembre de 2008 [Ref.- 130].
15. La evolución de las versiones posteriores de Android (desde las versiones 2.x a la versión 4.x) ha conllevado notables modificaciones sobre el kernel de Linux estándar, y pese a seguir empleándose versiones y referencias al kernel de Linux 3.x, en los años 2010 y 2011 cada vez el entorno Android distaba más de un entorno Linux tradicional. Sin embargo, ambos proyectos, Android y Linux, volvieron a reunificarse en la versión 3.3 del kernel².

¹ Imagen: http://www.techotopia.com/index.php/Image:Android_architecture2.png

² <http://www.zdnet.com/article/android-and-linux-re-merge-into-one-operating-system/>



16. Las versiones de Android (desde la versión 1.5) emplean nombres en clave correspondientes a nombres de postres y dulces en inglés, ordenados alfabéticamente. A continuación, se muestra el listado de versiones de Android conocidas en el momento de elaboración de la presente guía:

- *Alpha* (1.0)
 - *Beta* (1.1)
 - Cupcake (1.5)
 - Donut (1.6)
 - Eclair (2.0-2.1)
 - Froyo (2.2-2.2.3)
 - Gingerbread (2.3-2.3.7)
 - Honeycomb (3.0-3.2.6)
 - Ice Cream Sandwich (4.0-4.0.4)
 - Jelly Bean (4.1-4.3.1)
 - KitKat (4.4-4.4.4)
 - Lollipop (5.0)
 - Marshmallow (6.0)
 - Nougat (7.0)
17. Google mantiene una web en la que proporciona recomendaciones generales sobre buenas prácticas de seguridad para desarrolladores, que se

puede consultar para obtener una visión global de qué elementos resultan clave dentro de la plataforma Android para garantizar la seguridad del dispositivo móvil y las apps que en él se instalen [Ref.- 184].

4.1. ANDROID 2.X

18. El primer dispositivo móvil Google Nexus, basado en Android 2.1, fue el *smartphone* Nexus One (dispositivo empleado en la versión previa de la presente guía [Ref.- 401], con una pantalla de 3,7", y desarrollado por Google y HTC), que se empezó a comercializar en enero de 2010. Posteriormente, en mayo de 2010, fue actualizado a la versión Android 2.2 y comercializado como el primer dispositivo móvil con esta nueva versión de Android.
19. Android versión 2.2 (con nombre en clave "Froyo") fue la versión por defecto incluida en numerosos dispositivos móviles disponibles en el mercado durante los años 2010 y 2011, complementando versiones previas, como la versión 2.0/2.1 de Android ("Eclair"), 1.6 ("Donut") y 1.5 ("Cupcake") [Ref.- 130].
20. La versión 2.2 (analizada en la versión previa de la presente guía) añadió, entre otras, mejoras de rendimiento, la versión v8 del motor de JavaScript de Chrome, la posibilidad de actuar como hotspot Wi-Fi para compartir la conexión de datos de telefonía móvil con otros dispositivos, y soporte para Adobe Flash en el navegador web.
21. Android anunció una actualización en diciembre de 2010 asociada a la versión 2.3 ("Gingerbread") [Ref.- 9], que añadía mejoras en el interfaz de usuario, el teclado táctil y las capacidades de copiar y pegar datos, así como soporte para SIP (voz sobre IP, VoIP) y NFC (*Near Field Communication*), con capacidades NFC extras añadidas en la versión 2.3.3 [Ref.- 10]. El primer dispositivo móvil basado en Android 2.3 fue el *smartphone* Nexus S (desarrollado por Google y Samsung), comercializado en la misma fecha de lanzamiento que Android 2.3.

4.2. ANDROID 3.X

22. La versión 3.0 de Android ("Honeycomb") fue diseñada para dispositivos en formato tableta (o *tablets*), con pantalla de mayor tamaño y mayor capacidad de procesamiento (tanto del procesador principal como gráfico). Esta versión añadió la posibilidad de aplicar un cifrado completo a los datos almacenados en el dispositivo móvil [Ref.- 85]. El primer dispositivo móvil basado en Android 3.0, la tableta Motorola Xoom, se empezó a comercializar en el mercado en febrero de 2011.
23. Posteriormente se publicaron las versiones 3.1 y 3.2 en mayo y julio de 2011 respectivamente. La versión 3.1 añadió soporte para más dispositivos de entrada y conexión USB para la transferencia de ficheros, mientras que la versión 3.2 añadió soporte para tarjetas de memoria SD y mejoras en la gestión de la pantalla y de distintas resoluciones gráficas.

4.3. ANDROID 4.0

24. La versión 4.0 de Android ("Ice Cream Sandwich", ICS), basada en el kernel 3.0.1 de Linux, fue anunciada en octubre de 2011 y proporcionó gran parte de la funcionalidad existente en "Honeycomb" (Android 3.x) a los *smartphones* (no únicamente a *tablets*), añadiendo un gran número de funcionalidades y cambios en el interfaz de usuario, como por ejemplo mejoras en las capacidades multitarea, nuevas funcionalidades en la pantalla de desbloqueo (como el acceso directo a la cámara o el acceso al centro de notificaciones), control de acceso mediante reconocimiento facial utilizando la cámara frontal, control y monitorización del tráfico de red y datos (redes móviles y/o Wi-Fi), la posibilidad de compartir información mediante NFC (y Android Beam), soporte para Wi-Fi Direct (o Wi-Fi Peer-to-Peer, "P2P"), soporte para el perfil Bluetooth HDP (*Health Device Profile*) y para la conexión a sensores y dispositivos médicos y de salud inalámbricos, y nuevas capacidades de gestión y almacenamiento de credenciales y certificados digitales (KeyChain API [Ref.- 188]), entre otros. El primer dispositivo móvil basado en Android 4.0, el *smartphone* Galaxy Nexus (desarrollado por Google y Samsung), se empezó a comercializar en noviembre de 2011, y fue el primer Nexus con soporte para LTE (4G).
25. Android 4.0 implementó ASLR (*Address Space Layout Randomization*) para proteger tanto los componentes del sistema como las aplicaciones móviles (apps) de terceros frente a la explotación de vulnerabilidades debidas a una incorrecta gestión de la memoria. Adicionalmente, Android 4.0 incorporó el *Android VPN Framework* (AVF, o VPN API), que añade por defecto un cliente VPN con soporte para L2TP e IPsec. En versiones previas de Android el uso de software de conexión a redes VPN, en algunos escenarios, requería disponer de un dispositivo móvil Android *rooteado* (o con el proceso de *root* aplicado).
26. Desde el punto de vista de la gestión, se añadieron controles para poder deshabilitar remotamente la(s) cámara(s) del dispositivo móvil mediante las soluciones de gestión empresarial de dispositivos móviles o MDM (*Mobile Device Management*) y mediante el uso de la app de gestión de Google asociada (de tipo *Device Policy Manager*) en Android.

4.4. ANDROID 4.1

27. La versión 4.1 de Android ("Jelly Bean", JB) fue anunciada en la conferencia Google I/O en junio de 2012, centrada en mejorar la funcionalidad y rendimiento del interfaz de usuario, y con nuevas capacidades como las transferencias de datos mediante Bluetooth vía Android Beam (adicionalmente a las basadas en NFC), el descubrimiento de servicios a través de Wi-Fi Direct (o Wi-Fi "P2P"), o el descubrimiento de servicios a través de *multicast* DNS en redes Wi-Fi (Wi-Fi NDS, *Network Service Discovery*). El primer dispositivo móvil basado en Android 4.1, la tableta o *tablet* Nexus 7 (desarrollada por Google y Asus), se empezó a comercializar en julio de 2012.

28. Adicionalmente la versión 4.1 de Android añadió soporte para GCM (*Google Cloud Messaging*) [Ref.- 143], un servicio que permite el envío de mensajes y notificaciones *push* a los usuarios de apps específicas por parte de sus desarrolladores desde sus servidores a través de Google. Estas capacidades pueden ser también empleadas, por ejemplo, por las soluciones empresariales de gestión de dispositivos móviles (MDM).
29. Asimismo, Android 4.1 añadió múltiples capacidades a Google Play, como funcionalidad para proteger las apps de pago y sus contenidos mediante cifrado, empleando una contraseña específica para cada dispositivo móvil, antes de ser distribuidas y almacenadas desde la tienda o mercado oficial de apps de Google. Los nuevos servicios de Google Play también permiten integrar las capacidades de autenticación asociadas a la cuenta de usuario de Google en las apps de Android, así como integrarlas con Google+. Con el objetivo de mejorar y optimizar la distribución de nuevas actualizaciones de las apps, las actualizaciones de apps inteligentes (*Smart App Updates*) permiten enviar a los dispositivos móviles Android únicamente los elementos de la app que han sido actualizados, en lugar del paquete de Android (o APK) completo de la app.

4.5. ANDROID 4.2

30. La versión 4.2 de Android (también denominada "Jelly Bean"), basada en el kernel 3.4.0 de Linux, fue anunciada en octubre de 2012³, e introducía nuevas capacidades en la pantalla de desbloqueo (como el soporte para complementos, *widgets*), soporte para múltiples cuentas o perfiles de usuario (sólo para tabletas), la reimplementación de la pila NFC y de la pila Bluetooth (pasando de BlueZ a BlueDroid, esta última desarrollada por Google y Broadcom), soporte para pantallas externas (incluso a través de Wi-Fi, mediante *Wi-Fi Display*), la ocultación de las opciones de desarrollo por defecto, debiendo activarse éstas manualmente, y optimizaciones en el entorno de ejecución Dalvik. Los primeros dispositivos móviles basados en Android 4.2 fueron el Nexus 4 (*smartphone* desarrollado por Google y LG) y la Nexus 10 (tableta desarrollada por Google y Samsung), ambos comercializados en noviembre de 2012.
31. La versión 4.2 añadió notables mejoras de seguridad [Ref.- 140], destacando:
 - *Always-on VPN*, para permitir que las apps únicamente generen tráfico de red si existe una conexión VPN activa y proteger así el envío de datos no cifrados a través de otras redes que podrían acceder a la información transmitida (ej. directamente a través de Internet); ver apartado "19.6. Redes VPN".
 - Generación de notificaciones, y confirmación por parte del usuario, antes de realizar el envío de mensajes SMS *premium* (con coste adicional).
 - Verificación de apps, funcionalidad que permite analizar las aplicaciones móviles (apps) antes de su instalación para detectar si disponen de código

³ Android 4.2 fue simplemente anunciada a través de una nota de prensa debido al huracán Sandy.

o funcionalidades dañinas conocidas (*malware*); ver apartado "Verificación de seguridad de las apps: Google Play Protect" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407].

- La librería principal con soporte para SSL/TLS añadió soporte para *certificate pinning* [Ref.- 79], permitiendo disponer de un mayor control sobre el certificado digital empleado por una app o un dominio concreto, y evitando problemas de seguridad asociados a las autoridades certificadoras (CAs, *Certificate Authorities*).
 - Organización de los permisos de las apps en grupos, lo que simplifica su aprobación por parte del usuario estándar, pero limita la granularidad en su análisis y en la autorización por parte de usuarios avanzados.
 - El demonio *installd* encargado de la instalación de apps no ejecuta con permisos de root (o superusuario), limitando así la explotación de potenciales vulnerabilidades sobre este componente.
 - Los scripts de arranque (*init scripts*) aplican la semántica basada en la directiva `O_NOFOLLOW` para prevenir ataques mediante enlaces (o *links*) simbólicos.
 - Por defecto, las apps que hacen uso de la API 17 o superior no exportan sus proveedores de contenidos (ContentProviders), reduciendo así su exposición frente a posibles ataques.
 - Por defecto, las apps que hacen uso de la API 17 o superior y que utilizan WebViews no permiten el acceso directo al método "addJavascriptInterface()", limitando la ejecución remota de código en la interacción entre JavaScript y Java, y reduciendo así su exposición frente a posibles ataques.
 - Mejoras en la implementación por defecto para la generación de números aleatorios (SecureRandom) y RSA pasa a hacer uso de OpenSSL (1.0.1), con soporte para TLS v1.1 y v1.2.
 - Múltiples actualizaciones de seguridad de librerías, como por ejemplo WebKit, libpng, OpenSSL, y LibXML.
32. Adicionalmente, la versión Android 4.2 proporcionó una nueva funcionalidad para establecer conexiones USB seguras con un ordenador (concretamente desde la versión Android 4.2.2 y posteriores), autenticando las conexiones ADB mediante un par de claves RSA, y solucionó la vulnerabilidad Master Key (teóricamente desde Android 4.2.2, aunque la versión concreta depende del fabricante del dispositivo móvil).

4.6. ANDROID 4.3

33. La versión 4.3 de Android (también denominada "Jelly Bean"), fue liberada en julio de 2013, con soporte para Bluetooth Smart (o BLE, *Bluetooth Low Energy*, disponible inicialmente en los modelos Nexus 4 y Nexus 7 2013), acceso restringido para nuevos perfiles de usuario, la disponibilidad

temporal de "App Ops" (un sistema granular de control de permisos para las aplicaciones móviles, oculto por defecto⁴), optimizaciones en el cálculo de la localización vía hardware en lugar de software (*hardware geofencing*) y soporte para la obtención de información de redes Wi-Fi cercanas para los servicios de localización, incluso cuando el interfaz Wi-Fi está apagado. La mayoría de dispositivos móviles Nexus obtuvieron la actualización 4.3 a la semana de su publicación, y el primer dispositivo móvil en ser comercializado directamente con Android 4.3 fue la segunda generación de la *tablet* Nexus 7, conocida como Nexus 7 2013 (también desarrollada por Google y Asus), en julio de 2013.

34. La versión 4.3 añadió notables mejoras de seguridad [Ref.- 140], destacando:

- Las apps pueden configurar las credenciales de redes Wi-Fi WPA(2)-Empresariales que necesiten, pudiendo emplear múltiples métodos de autenticación basados en el estándar EAP (*Extensible Authentication Protocol*).
- La API de KeyChain dispone de nuevas capacidades para establecer que un conjunto de credenciales no puede ser exportado fuera del dispositivo móvil, es decir, se vinculan al hardware o dispositivo móvil concreto. Asimismo, las apps disponen de capacidades para almacenar credenciales en un Keystore privado que no esté accesible a otras apps, y estas credenciales pueden ser añadidas sin interacción del usuario, siendo posible definir que tampoco puedan ser exportadas.
- La partición de sistema ("/system") en Android 4.3 es montada con restricciones mediante el parámetro o flag *nosuid*, y los procesos derivados de zygote no disponen de capacidades para ejecutar programas SUID, limitando por tanto la ejecución de programas con *setuid* o *setgid* por parte de las apps (por ejemplo, con privilegios de root o superusuario), y evitando así la explotación de ciertas vulnerabilidades previamente conocidas. Adicionalmente se eliminaron numerosos programas *setuid* o *setgid*.
- Tanto zygote como ADB reducen sus capacidades antes de ejecutar apps, evitando así que las apps ejecutadas tanto desde el propio Android como desde una *shell* (a través de ADB) no dispongan de capacidades privilegiadas. Adicionalmente zygote evita que se añadan nuevas capacidades antes de ejecutar código por parte de las apps, evitando la elevación de privilegios vía "execve".
- Utilización de las capacidades FORTIFY_SOURCE por parte de las librerías y aplicaciones del sistema para prevenir la explotación de vulnerabilidades de *strings* no terminados adecuadamente o de corrupción de memoria, junto a otras protecciones en la reubicación de código (*relocation*) para binarios enlazados estáticamente y en el propio código de Android.

⁴ Este sistema granular de control de permisos para las aplicaciones móviles fue eliminado en una versión posterior de Android por Google, concretamente en la versión 4.4.2. El mismo actuó como el precursor del modelo de permisos en tiempo de ejecución introducido en Android 6.x.

- Soporte para MAC (*Mandatory Access Control*, frente a *Discretionary Access Control*, DAC) en Android a través de SELinux (Security-Enhanced Linux) [Ref.- 142] y la definición asociada de políticas de seguridad. En Android 4.3 se hace uso de SELinux en modo permisivo o *permissive mode*, donde únicamente las acciones no legítimas son registradas por el kernel, pero no bloqueadas [Ref.- 141].
- Las capacidades MAC (*Mandatory Access Control*) de SELinux en Android se pueden emplear en todos los procesos, tanto los asociados a las apps como los procesos de sistema que ejecutan como root o system (aplicando el modelo de permisos basado en *Linux capabilities*), dependiendo de la versión de Android.

4.7. ANDROID 4.4

35. La versión 4.4 de Android ("KitKat", KK [Ref.- 148]) fue anunciada en septiembre de 2013, optimizada específicamente para ejecutar en dispositivos más básicos con capacidades reducidas, incluso con tan sólo 512 MB de memoria RAM (o incluso menos⁵; *Project Svelte*⁶). Adicionalmente, esta versión añadió capacidades de impresión inalámbricas, un nuevo proveedor de SMS para la gestión de mensajes (SMS/MMS), soporte para emulación de *host* en NFC (HCE, *Host Card Emulation*), permitiendo que las apps de un dispositivo móvil lean y emulen *smartcards* NFC (como las empleadas como medio de pago en las tarjetas EMV), incorporó nuevos perfiles Bluetooth (*Bluetooth HID over GATT* (HOGP), MAP (*Message Access Profile*) y extensiones para AVRCP), soporte para *Wi-Fi Tunneled Direct Link Setup* (TDLS), añadió una implementación nueva de WebView basada en Chromium, añadió también la certificación *Wi-Fi Display Specification* haciendo compatibles con Miracast a los dispositivos móviles Android 4.4 (como por ejemplo el modelo Nexus 5), e introdujo el nuevo entorno de ejecución ART (*Android RunTime*) como futuro sustituto de Dalvik, pero todavía de manera experimental y no habilitado por defecto. El primer dispositivo móvil basado en Android 4.4 fue el *smartphone* Nexus 5 (desarrollado por Google y LG), comercializado en octubre de 2013. Posteriormente, la actualización de Android 4.4 (KitKat) se extendió a otros dispositivos móviles de Google, como el Nexus 4, Nexus 7 o Nexus 10, y de otros fabricantes.

Nota: Android 4.4 (KitKat) introdujo soporte nativo para (Google) Cloud Print⁷, un servicio que permite imprimir desde cualquier dispositivo móvil Android (y otros dispositivos y ordenadores), y desde cualquier lugar, en cualquier impresora conectada a Cloud Print, y cuya configuración está disponible bajo el menú "Ajustes [Sistema] - Impresión". Las versiones previas de Android pueden utilizar la app Cloud Print⁸,

⁵ El valor mínimo de RAM necesario para ejecutar Android 4.4.x se supone que es de 340MB.

⁶ <http://android-developers.blogspot.com.es/2013/10/android-44-kitkat-and-updated-developer.html>

⁷ <http://www.google.com/cloudprint/learn/index.html>

⁸ <https://play.google.com/store/apps/details?id=com.google.android.apps.cloudprint&hl=es>

disponible en Google Play, para hacer uso de este servicio. Se recomienda deshabilitar las capacidades de impresión de Android si no se está haciendo uso de ellas (habilitadas por defecto). El análisis de seguridad de Cloud Print queda fuera del alcance de la presente guía.

Igualmente, Android dispone de capacidades para extender la funcionalidad del teclado existente por defecto mediante la instalación de apps de teclado de terceros (IME, *Input Method Editor*). Debido a que el teclado gestiona la mayoría de la entrada de datos del usuario, incluyendo contraseñas, un teclado malicioso podría registrar todo lo que teclea el usuario en un fichero local, e incluso enviarlo a un servidor remoto a través de Internet. Por este motivo, debe prestarse especial atención a los teclados actualmente instalados, disponibles desde el menú "Ajustes [Personal] - Idioma e introducción de texto [Teclado y métodos de introducción]", eliminando aquellos en los que no se confíe, y tomar precauciones a la hora de instalar nuevos teclados. El análisis de seguridad de teclados adicionales en Android queda fuera del alcance de la presente guía.

36. Las subversiones de la versión 4.4 añadieron capacidades destacables, como una mejor compatibilidad con las aplicaciones móviles para ART en 4.4.1, la eliminación del sistema de control de permisos "App Ops" en 4.4.2 (añadido en Android 4.3) o cambios para solucionar la vulnerabilidad de Heartbleed de OpenSSL en 4.4.3.
37. La versión 4.4 añadió notables mejoras de seguridad [Ref.- 86][Ref.- 140], destacando:
 - Granularidad para configurar redes VPN por usuario en dispositivos multi-usuario, permitiendo a un usuario redirigir todo su tráfico a través de una red VPN sin afectar a otros usuarios.
 - Soporte para las capacidades FORTIFY_SOURCE de nivel 2, y su aplicación a todo el código fuente de Android, que ha sido compilado con estas protecciones.
 - Soporte para dos nuevos algoritmos criptográficos: *Elliptic Curve Digital Signature Algorithm* (ECDSA y DSA) para el almacenamiento de credenciales en los proveedores del Keystore y la utilización de firmas digitales, y soporte para la función de derivación de claves con scrypt a la hora de proteger las claves empleadas en el cifrado completo del dispositivo móvil.
 - Notificaciones de seguridad avisando al usuario cuando se añaden certificados digitales al repositorio del dispositivo móvil.
 - Capacidades de protección adicionales empleando *certificate pinning* para detectar y prevenir el uso de certificados fraudulentos en comunicaciones SSL/TLS asociadas a los servicios de Google.
 - Mejoras para eliminar las vulnerabilidades PileUp o Fake ID (existente desde Android 2.1).
 - En Android 4.4 se hace uso de SELinux en modo obligatorio o *enforcing mode* parcialmente, es decir, sólo para un conjunto crítico de procesos o

dominios de sistema (installd, netd, vold y zygote), en el que todas las acciones no legítimas son bloqueadas y registradas por el kernel [Ref.- 141] [Ref.- 142]. Para el resto de procesos o dominios, SELinux en Android 4.4 sigue actuando en modo permisivo, como en el caso de Android 4.3.

38. La versión 4.4.4 de Android soluciona específicamente una vulnerabilidad de seguridad asociada a OpenSSL que permitía la realización de ataques MitM (Man-in-the-Middle) tanto sobre clientes como sobre servidores, con el identificador CVE-2014-0224⁹, y que permite forzar el uso de una clave maestra débil (vacía) en ciertos escenarios de comunicación entre versiones de OpenSSL, conocida como la vulnerabilidad de "inyección CCS (ChangeCipherSpec)" [Ref.- 139] [Ref.- 140]¹⁰.
39. Posteriormente, en julio de 2014, y asociada a la API 20 (en lugar de la API 19 de la versión 4.4 de Android estándar) se anunció la versión 4.4W (o 4.4WE) de Android, con extensiones para dispositivos personales (WE, *Wearable Extensions*) [Ref.- 136].

Nota: El resumen previo de las diferentes versiones de Android no hace referencia específica a las subversiones, como por ejemplo las versiones 4.0.4, 4.3.1 o 4.4.4, salvo que se haya considerado relevante destacar alguna funcionalidad concreta de alguna subversión.

Como referencia, se han incluido referencias a los dispositivos móviles de la gama Google Nexus¹¹ asociados a cada versión principal de Android.

4.8. ANDROID 5.X

40. La versión 5.0 de Android ("Lollipop", L) fue liberada en octubre de 2014 (tras su anuncio inicial o *preview* en junio de 2014, durante la conferencia Google I/O), con notables mejoras incluyendo el nuevo interfaz gráfico de usuario basado en *Material Design*, haciendo uso de ART (*Android RunTime*) por defecto (y sus mejoras de rendimiento asociadas), con soporte para arquitecturas y CPUs de 64 bits, mejoras en el ahorro de batería con el proyecto Volta¹², nuevas capacidades asociadas a la pantalla de desbloqueo y al centro y capacidades de notificaciones, mejoras en Bluetooth Low Energy (BLE) y NFC para realizar operaciones concurrentes en Bluetooth (escaneos y anuncios¹³) y que las apps registren su identificador de app NFC (AID) dinámicamente, estadísticas de uso de las apps, un nuevo espacio denominado "Overview" (previamente denominado "Recientes") para la gestión de las tareas y apps en ejecución, mejoras en los gráficos (con

⁹ <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224>

¹⁰ Esta vulnerabilidad se solucionó adicionalmente a través de los servicios de Google Play (GPS, *Google Play Services*) para otras versiones de Android previas.

¹¹ http://en.wikipedia.org/wiki/Google_Nexus y <http://www.google.es/nexus/>.

¹² Herramientas: "dumpsys batterystats" y Battery Historian (<https://github.com/google/battery-historian>).

¹³ Escaneos (modo central) y anuncios (modo periférico).

soporte para OpenGL ES 3.1), las capacidades multimedia, el audio y la cámara (con soporte más granular para la captura de fotos y el procesamiento de imágenes), la posibilidad por parte del usuario de evitar interrupciones externas durante un periodo de tiempo definido, disponibilidad de un nuevo sensor de inclinación para detectar la actividad del usuario y un sensor del ritmo cardíaco, etc. [Ref.- 500]

41. Las apps en Android 5.0 disponen de la capacidad (tras obtener autorización del usuario) de capturar la pantalla del dispositivo y compartirla a través de la red.
42. Android 5.x facilita el cambio del idioma, en concreto del IME (*Input Method Editor*), por parte del usuario desde el icono con forma de globo terráqueo del teclado.
43. Asimismo, Android 5.x introduce capacidades para que una app pueda acceder a los contenidos multimedia de otra app, que previamente debe haber exportado los mismos a través de un servicio (*MediaBrowserService*) [Ref.- 502].
44. El acceso a las capacidades de almacenamiento también se extiende en Android 5.x, permitiéndose la selección por parte de las apps de un directorio y de todos sus contenidos, y subdirectorios, en una sola operación, en lugar de individualmente.
45. El soporte para arquitecturas de 64 bits de Android 5.x, aun manteniendo el soporte completo de 32 bits, incluye un mayor espacio de direccionamiento, mejoras de rendimiento generales, en particular para las operaciones criptográficas de OpenSSL, junto a las capacidades asociadas del NDK (*Native Development Kit*), revisión 10c [Ref.- 503] [Ref.- 504].
46. Android 5.x tiene como objetivo ofrecer un mayor tiempo de uso mediante la optimización de la batería y un menor tiempo de carga. Se introduce una función de ahorro de energía, que puede alargar la duración de la batería restante hasta 90 minutos. Asimismo, Android 5.x permite consultar el tiempo estimado que queda para que se agote la batería y el tiempo estimado restante para que finalice la carga de la batería durante el proceso de carga.
47. En versiones previas de Android, la pantalla de "Recientes" o lista de tareas en ejecución sólo mostraba una tarea por cada app ejecutada recientemente, mientras que la nueva pantalla de visualización general de tareas en ejecución de Android 5.x permite disponer de más de una tarea por app, facilitando la multitarea y la interacción con actividades y documentos de diferentes apps de manera consistente [Ref.- 502], como por ejemplo, mostrando distintas tareas para las diferentes pestañas abiertas del navegador web.
48. En Android 5.0 se hace uso de SELinux en modo obligatorio o *enforcing mode* globalmente o completo (*full enforcement*), es decir, tanto para el conjunto crítico de procesos de sistema (ya protegidos en Android 4.4), como para el resto de procesos (con más de 60 dominios) [Ref.- 141][Ref.- 142].

49. Los primeros dispositivos móviles basados en Android 5.x fueron el *smartphone* Nexus 6, con una pantalla de 6" (desarrollado por Google y Motorola) y la *tablet* Nexus 9 (desarrollada por Google y HTC), comercializados en octubre/noviembre de 2014, junto al Nexus Player. Posteriormente, en noviembre de 2014 la actualización de Android 5.x (Lollipop) se extendió a otros dispositivos móviles de Google, como el Nexus 4, 5, 7 y 10, y a dispositivos móviles de otros fabricantes.
50. Posteriormente, en diciembre de 2014 se liberaron las versiones 5.0.1 y 5.0.2 de Android, con mejoras de funcionalidad y errores. Asimismo, en marzo de 2015 se liberó la versión 5.1, que permitía conectarse a redes Wi-Fi y/o emparejarse con otros dispositivos Bluetooth directamente desde los ajustes de configuración rápidos de Android, incluía soporte oficial para múltiples tarjetas SIM [Ref.- 509] y capacidades para aprovisionar los ajustes de los operadores móviles de telecomunicaciones. La versión 5.1 habilitaba por defecto "Factory Reset Protection" (FRP), un mecanismo de seguridad para evitar que un dispositivo móvil Android que haya sido robado pueda ser reutilizado, ya que, para su activación, incluso tras restaurar el mismo a los ajustes por defecto de fábrica, es necesario disponer de las credenciales del usuario de Google de su último propietario. La versión 5.1.1 liberada en abril de 2015 ofrecía de nuevo corrección de errores.
51. En concreto, Android 5.1 añade soporte para hacer uso de múltiples tarjetas SIM simultáneamente (en aquellos dispositivos móviles que disponen de varias ranuras), elimina el soporte para componentes antiguos del SDK empleados en las comunicaciones HTTP, dando prioridad a URLConnection, y proporciona soporte para que los proveedores de telecomunicaciones puedan desarrollar apps que puedan modificar los ajustes de configuración del operador móvil y llevar a cabo otras tareas de aprovisionamiento. Este tipo de apps deben de estar firmadas por un certificado de desarrollador que coincida con el certificado UICC (Universal Integrated Circuit Card) del dispositivo móvil [Ref.- 511].
52. Una mejora sustancial de la versión 5.1 fue la resolución del "*memory leak*" que introdujo Android 5.0 y que provocaba que la memoria del dispositivo se llenara, causando fallos en las aplicaciones que ejecutan en segundo plano y en la propia pantalla de inicio. Aunque no afectó a todos los terminales, sí fue un problema muy extendido, especialmente entre aquellos dispositivos con memoria RAM inferior a 2 GB [Ref.- 510].
53. Android 5.1 deja de intentar conectarse automáticamente a aquellas redes Wi-Fi que no proporcionan acceso a Internet. Si una red Wi-Fi que había sido guardada en el dispositivo móvil deja de proporcionar acceso a Internet, aparecerá un mensaje justo debajo del nombre de la red Wi-Fi informando de esa circunstancia.

4.9. ANDROID 6.X

54. La versión 6.x de Android ("Marshmallow", M) fue liberada en octubre de 2015 (tras su anunció inicial o *preview* en mayo de 2015, durante la

conferencia Google I/O), con soporte para los dispositivos móviles Nexus 5, 6, 7 (sólo para la versión de 2013) y 9. Esta versión incluía mejoras adicionales para el ahorro de batería a través del proyecto Doze, capacidades "Direct Share" para la compartición de datos directamente entre apps¹⁴, optimizaciones de contenido para el "Asistente"¹⁵, soporte para los nuevos conectores USB de tipo C (con capacidad de carga ultrarrápida), capacidades automáticas para la realización de copias de seguridad (o backups) automáticas de los datos de las apps (y su recuperación), *verified boot* (ver apartado "7.9. Proceso de arranque seguro: Verified Boot"), compatibilidad con dispositivos MIDI, la introducción del concepto de dispositivos de almacenamiento flexibles o adoptables [Ref.- 603] (ver apartado "10.3. Dispositivos de almacenamiento"), capacidades que facilitan la transferencia de datos, cuentas y apps durante el proceso de migración a un nuevo dispositivo móvil, funcionalidad para añadir una cuenta de e-mail adicional (personal o corporativa) durante el proceso de instalación y configuración inicial, soporte para el estándar Wi-Fi Hotspot 2.0¹⁶ y para la creación de zonas Wi-Fi en las bandas de frecuencia de 5 GHz, soporte para el perfil SAP (*SIM Access Profile*) de Bluetooth que permite hacer llamadas de teléfono desde un vehículo empleando la tarjeta SIM del dispositivo móvil, numerosas nuevas capacidades en Android for Work (ver apartado "8.2. Android for Work"), etc.

55. Android 6.x introduce de forma oficial el "Administrador de permisos", una funcionalidad que permite al usuario gestionar a nivel de aplicación los permisos de calendario, contactos, cámara, micrófono, SMS, sensores, teléfono y ubicación (ver apartado "7.1. Modelo de permisos de Android").
56. Android 6.x introduce la función "Now on tap", cuyo propósito es proporcionar al usuario información contextualizada sobre lo que muestra la aplicación que se encuentra en primer plano cuando se invoca "Now on tap" (ver apartado "Asistente de Google, Mi tablón y servicios de búsqueda de Google" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]).
57. En relación con el punto anterior, Android 6.x proporciona una integración más directa con el servicio denominado *feed* (cuyo precursor es Google Now), de forma que, manteniendo pulsado el botón de inicio, ofrece más capacidades a la hora de interactuar con el dispositivo móvil y sus servicios asociados a través de la voz [Ref.- 600].
58. En concreto, Android 6.x facilita a las aplicaciones móviles nuevas capacidades para interactuar y llevar a cabo acciones por parte del usuario mediante la voz [Ref.- 603], así como facilitar la interacción con las apps a través del Asistente de Google, pudiendo éste disponer del contexto

¹⁴ <https://developer.android.com/about/versions/marshmallow/android-6.0.html#direct-share> [Ref.- 603]

¹⁵ <https://developer.android.com/training/articles/assistant.html>

¹⁶ Soporte para la especificación Hotspot 2.0 Release 1 en dispositivos Nexus 6 y Nexus 9 [Ref.- 603].

asociado a cada app. El "Asistente" es activado mediante una pulsación prolongada del botón de inicio (o *Home*).

59. Android 6.x introdujo soporte oficial para los sensores de huellas dactilares digitales (*fingerprint*), junto a las capacidades para hacer uso de estos durante el proceso de autenticación del usuario (o desbloqueo del dispositivo móvil) y su integración para la realización de pagos dentro de las aplicaciones móviles (ver apartado "9.3. Huella dactilar digital (*fingerprint*)").
60. Android 6.x añade soporte completo para memorias microSD y USB, de forma que es posible conectar mediante USB (haciendo uso de un cable USB OTG, *On-The-Go*) un dispositivo de almacenamiento externo para acceder y ver archivos multimedia o para usarlo como medio de almacenamiento externo (ver apartado "10.2. Almacenamiento adoptable").
61. Android 6.x introduce un nuevo control de audio, de forma que, al pulsar sobre el botón de volumen, se desplegará un control de sonido que permite regular los controles de volumen del tono, multimedia y de las alarmas, incluso con el dispositivo bloqueado.
62. Android 6.x introduce el denominado "Configurador de IU (interfaz de usuario) del sistema", una opción experimental que habilita determinadas funcionalidades para personalizar el dispositivo móvil desde el centro de control, disponible en la parte superior, desde los ajustes de configuración rápidos (ver apartado "9.10. Configurador de IU del sistema").
63. Por otro lado, las optimizaciones de la batería incluyen la función de "Aplicaciones en Espera" (o apps inactivas, limitando el consumo por parte de apps que apenas se utilizan) y un modo "Descanso" (estado suspendido) para que el consumo sea mínimo cuando no se hace uso del dispositivo móvil y que, por ejemplo, pueda permanecer toda la noche encendido para hacer sonar una alarma a la mañana siguiente.
64. El modo "No molestar" (ver apartados "9.7. Interrupciones" y "9.8. Notificaciones") puede ser gestionado fácilmente desde los ajustes de configuración rápidos, y configurado para que una llamada de un mismo remitente en un periodo de 15 minutos suene de manera estándar. Este modo puede ser activado automáticamente para ciertos periodos de tiempo o para ciertos eventos del calendario.
65. La funcionalidad de realización de copias de seguridad (backups), y restauración, de Android 6.x incluye nuevos ajustes de configuración del sistema, como por ejemplo los ajustes de sincronización, las apps preferidas, la configuración del modo "No molestar", los ajustes de accesibilidad y los métodos de introducción de texto previamente habilitados.
66. Los nuevos ajustes simplificados, disponibles desde "Ajustes - [Dispositivo] Aplicaciones", permiten gestionar todas las características y configuraciones de las apps desde una misma ubicación, incluyendo el consumo de batería o memoria, sus permisos y/o notificaciones.
67. En Android 6.x, se cambió de nuevo el comportamiento de la pantalla de "Recientes" para el navegador web nativo (Chrome), de forma que, en lugar

de mostrarse cada página web abierta en el navegador de forma independiente (como una tarea separada), sólo se muestra una única tarea, siendo necesario abrir la app del navegador web y pulsar sobre el icono que representa el número de pestañas abiertas para acceder a la lista completa de las mismas.

68. Android 6.x introdujo el mecanismo "Direct Share", a través del cual una aplicación móvil puede exponer parte de su información y datos a otras aplicaciones para que éstas la utilicen (ver apartado "7.4. Compartición de datos entre apps y Direct Share").
69. Android 6.x introdujo el denominado "Doze", un sistema de ahorro de batería similar al modo hibernación de los ordenadores personales (ver apartado "9.9. Ahorro de batería").
70. Android 6.x elimina el soporte para el cliente HTTP de Apache, promoviendo el uso de `HttpURLConnection` en su lugar para apps con un nivel de API igual o superior a 9 (Android 2.3). Esta clase hace un uso más eficiente de las conexiones HTTP(S) a través de mecanismos de compresión, transparentes para el usuario, y de caché [Ref.- 602].
71. Android está dejando a un lado el uso de OpenSSL como librería criptográfica, y haciendo uso en su lugar de BoringSSL [Ref.- 605]. En el caso de hacer uso del NDK, no se deben referenciar directamente librerías que no pertenezcan a la API del NDK, como por ejemplo "libcrypto.so" y "libssl.so" (librerías no públicas que pueden cambiar sin previo aviso), y en su lugar, se debería hacer uso de las APIs criptográficas de Java a través de JNI, *Java Native Interface* [Ref.- 602].
72. Android 6.x bloquea el acceso directo a los identificadores hardware del dispositivo móvil, como, por ejemplo, las direcciones MAC asociadas a las APIs de Wi-Fi y Bluetooth [Ref.- 602].
73. Por otro lado, en Android 6.x o superior, para que una app pueda acceder a las direcciones MAC de otros dispositivos externos cercanos a través de escaneos Bluetooth y Wi-Fi, debe solicitar los permisos `ACCESS_FINE_LOCATION` o `ACCESS_COARSE_LOCATION` [Ref.- 602].
74. Los escaneos en *background* o de fondo (Wi-Fi y Bluetooth) se llevan a cabo empleando una dirección MAC origen aleatoria, por lo que los dispositivos externos no pueden identificar de manera directa la dirección MAC real del dispositivo móvil hasta que establezcan una conexión con él.
75. En Android 6.x las apps no disponen de acceso global a la lista de webs favoritas (o *bookmarks*) del usuario, habiéndose eliminado los permisos existentes previamente para este acceso, `READ_HISTORY_BOOKMARKS` y `WRITE_HISTORY_BOOKMARKS`. En su lugar, las apps con un nivel de API objetivo igual o superior a 23 deben gestionar internamente su propia lista de web favoritas [Ref.- 602].
76. La implementación de Keystore de Android 6.x no soporta DSA, debiéndose emplear ECDSA (Elliptic Curve DSA) en su lugar [Ref.- 602].

77. Las claves que no requieren ser almacenadas en formato cifrado, no serán eliminadas cuando se deshabilita el uso de un código de acceso en la pantalla de desbloqueo, por parte del usuario o de una app administradora del dispositivo. Las que sí solicitan su almacenamiento cifrado sí serán eliminadas en ese escenario.
78. El entorno de ejecución de Android 6.x no permite a las apps realizar la carga de librerías dinámicas o compartidas con capacidad de reubicar sus secciones de texto (*text relocations*) [Ref.- 602], con el objetivo de disponer de código independiente de su posición en memoria (PIC, *Position Independent Code* o PIE, *Position Independent Executable*) y poder así hacer uso de ASLR. En versiones previas de Android, al llamar a la función "dlopen()" con este tipo de librerías se generaba un aviso, pero sí se permitía la carga de la librería.
79. En Android 6.x se lleva a cabo una verificación más estricta de los ficheros APK a la hora de instalar una app, considerándose que el fichero APK está corrupto si un fichero declarado en el fichero *manifest* no está presente dentro del fichero APK. Por tanto, en caso de que cualquier contenido sea eliminado del fichero APK, será necesario firmar de nuevo dicho fichero [Ref.- 602].
80. Android 6.x proporciona nuevos modos de conexión a través del puerto USB del dispositivo móvil, obligando a que la transferencia de datos por USB sea iniciada a través de la notificación que se muestra cuando se conecta otro dispositivo y no desde el menú de "Ajustes [Dispositivo] - Almacenamiento & USB - [...] - Conexión USB a ordenador", tal como ocurría en versiones previas de Android (ver apartado "13. Comunicaciones USB").
81. Los primeros dispositivos móviles basados en Android 6.x fueron los *smartphones* Nexus 5X y 6P (desarrollados por Google junto a LG y Huawei, respectivamente), y comercializados en octubre de 2015, ambos con sensor de huella dactilar digital en su parte posterior y con el nuevo conector USB de tipo C.
82. Desde el punto de vista de seguridad, cabe destacar (tal como se ha mencionado anteriormente) la introducción en Android 6.x de un *framework* o entorno (estándar en dispositivos móviles compatibles y con un lector de huella o *fingerprint*) para llevar a cabo la autenticación mediante huella dactilar digital, que, no sólo permite el desbloqueo del dispositivo móvil (ver apartado "9.3. Huella dactilar digital (*fingerprint*)"), sino también la autenticación desde las apps mediante huella (o incluso con otras credenciales de la plataforma móvil mediante la API Confirm Credential), y la realización de compras mediante Android Pay y de compras en Google Play, junto al inicio de sesión en esta última plataforma.
83. Por otro lado, también desde el punto de vista de seguridad es necesario destacar que Android 6.x introdujo un nuevo modelo de permisos en tiempo de ejecución para las apps, donde ya no es necesario que el usuario otorgue a la app todos los permisos solicitados en el momento de su instalación, sino que estos pueden ser otorgados posteriormente y de manera granular y

selectiva (ver apartado "7.1.2. Modelo de Permisos en tiempo de ejecución"). Asimismo, Android 6.x proporciona controles para gestionar (activar y desactivar) individualmente los permisos de cada app, o de las apps que tienen acceso a un grupo concreto de permisos.

84. En diciembre de 2015 se liberó la versión 6.0.1 de Android, con pequeñas mejoras de funcionalidad [Ref.- 86].

4.10. ANDROID 7.X

85. La versión 7.x de Android ("Nougat", o Android N) fue liberada por Google a finales de agosto de 2016¹⁷, con soporte para los dispositivos móviles Nexus 6, 9, 5X y 6P, y significativos cambios ofrecidos a través de su API de nivel 24 [Ref.- 703].
86. Esta versión incluye mejoras mediante un entorno multi-ventana¹⁸ o de pantalla dividida, para poder ejecutar, visualizar e interactuar con dos apps simultáneamente, la capacidad de cambiar rápidamente entre las dos apps recientemente usadas mediante una doble pulsación, la opción de responder directamente a las notificaciones sin abrir ninguna app o de recibir notificaciones agrupadas por app¹⁹, nuevas capacidades de realidad virtual (VR, *Virtual Reality*) mediante Google Daydream²⁰ (en los dispositivos móviles compatibles), integración de Vulkan (un estándar abierto para gráficos 3D y renderización), capacidades para limitar el consumo de datos (2/3/4G) de las apps que ejecutan de fondo con Data Saver, o soporte multi-idioma, entre otros. Por otro lado, el sistema de backup (o copia de seguridad) ha sido mejorado para incluir nuevos elementos, como, por ejemplo, los permisos en tiempo de ejecución de las apps, los ajustes de *hotspots* Wi-Fi o las restricciones de redes Wi-Fi. Asimismo, Android 7.x incluye nuevas capacidades para el ahorro de batería, denominadas "Doze on the Go", cuando el dispositivo móvil se encuentra en el bolsillo, maleta o bolso del usuario [Ref.- 700].
87. Una de las características más llamativas de Android 7.x es el cambio en el interfaz de la app "Ajustes", a través de la cual se accede a la mayor parte de las opciones de configuración del dispositivo móvil. El nuevo interfaz de acceso a los ajustes se describe en el apartado "9. Acceso físico al dispositivo móvil - Interfaz de usuario".
88. Android 7.x permite disponer de dos apps en ejecución en la pantalla del dispositivo móvil, pudiendo redimensionarse el espacio que ocupa cada una de ellas [Ref.- 702]. Esta funcionalidad permite a las apps hacer uso de capacidades de arrastrar y soltar para compartir contenidos fácilmente entre ellas.

¹⁷ <https://android-developers.blogspot.com.es/2016/08/taking-final-wrapper-off-of-nougat.html>

¹⁸ <https://android-developers.blogspot.com/2016/05/designing-for-multi-window.html>

¹⁹ <https://android-developers.blogspot.com.es/2016/06/notifications-in-android-n.html>

²⁰ https://vr.google.com/intl/en_US/daydream/

89. Las nuevas y numerosas capacidades asociadas a las notificaciones en Android 7.x hacen que prácticamente el nombre, Nougat o N, esté vinculado a las "Notificaciones". Se han añadido capacidades de personalización para hacer uso de plantillas (incluyendo avatares), cambiar el estilo de los mensajes (incluido el título y los contenidos) y su apariencia, agrupando notificaciones en base a la app o al tema de los mensajes, permitir al usuario actuar sobre el grupo o conjunto completo de notificaciones, responder directamente a las notificaciones sin abrir ninguna app desde el propio interfaz de las notificaciones, o emplear vistas personalizadas para modificar las cabeceras y acciones asociadas a las notificaciones [Ref.- 702]. Asimismo, añade capacidades al configurador de IU del sistema relativas a las notificaciones (ver apartado "9.8. Notificaciones").
90. Android 5.x incluyó el servicio "Notification Ranking Service"²¹, cuyo objeto es agrupar las notificaciones por importancia, en base a cómo de recientes sean, la app asociada (por ejemplo, teniendo prioridad las apps de mensajería instantánea), o el contacto. Estas capacidades han sido extendidas en Android 7.x.
91. La presentación de notificaciones se ha rediseñado, reemplazando las tarjetas (cards) por hojas (sheets) por aplicación, permitiendo acceso instantáneo a determinados ajustes, como bloquear o silenciar las notificaciones de una determinada app.
92. El primer dispositivo móvil basado en Android 7.x fue el *smartphone* LG V20 [Ref.- 700] (desarrollado por Google y LG), y comercializado inicialmente en Corea del Sur en septiembre de 2016, y en otros países en octubre de 2016. Adicionalmente, los otros dos dispositivos móviles basados en Android 7.x, diseñados y comercializados directamente por Google (aunque fabricados junto a HTC) son el Pixel y el Pixel XL [Ref.- 704], liberados en octubre de 2016, descontinuando así la gama de los dispositivos móviles Google Nexus, tras el Nexus 5X y 6P después de muchos años (desde 2010 con el Nexus One inicial) [Ref.- 705].
93. Android 7.x sustituyó su JRE (Java Runtime Environment) basado en Apache Harmony por OpenJDK. Su entorno de ejecución ART (Android Runtime) introduce mejoras de rendimiento y un compilador JIT (*Just in Time*) con capacidades de análisis del código (*profiling*). Esta característica permite optimizar y mejorar el rendimiento de las apps según son ejecutadas. El nuevo compilador JIT de ART complementa al compilador AOT (*Ahead of Time*), reduce el espacio de almacenamiento necesario para las apps, y acelera tanto la instalación de actualizaciones de apps como de sistema operativo [Ref.- 702]. La instalación de nuevas apps, y las actualizaciones de apps y de sistema, se lleva a cabo de manera inmediata y mucho más rápidamente, posponiendo las tareas de optimización.

²¹

<https://android.googlesource.com/platform/frameworks/base/+0fa02c7/packages/ExtServices/src/android/ext/services/notification/Ranker.java>

94. El análisis del código de las apps permite optimizar especialmente los métodos y módulos clave más utilizados en cada app, y reduce sus requisitos de memoria RAM. Por otro lado, las tareas de precompilación de AOT intentan minimizar el consumo de batería, llevándose a cabo cuando el dispositivo móvil está ocioso y conectado a la corriente eléctrica.
95. Android 6.x introdujo el denominado "Doze" para optimizar el consumo de batería cuando el dispositivo móvil no está siendo usado y permanece inmóvil, por ejemplo, encima de una mesa [Ref.- 616]; Android 7.x mejora dichas capacidades con "Doze on the Go", aplicando optimizaciones al consumo incluso cuando el dispositivo móvil está en movimiento pero no activo (p.ej. en el bolsillo del usuario) [Ref.- 702].
96. En Android 7.x se han mejorado las optimizaciones de memoria RAM que comenzaron con el proyecto *Svelte* en Android 4.4 (ver apartado "4.7. Android 4.4"), centrándose en cómo ejecutan las apps de fondo (o *background*) [Ref.- 702].
97. Android 7.x incluye también Data Saver, una funcionalidad disponible a través de "Ajustes - Uso de datos - Ahorro de datos" que permite limitar el consumo de datos móviles (2/3/4G) de las apps que ejecutan de fondo [Ref.- 702]. Estas capacidades son especialmente útiles cuando el dispositivo móvil se encuentra en el extranjero haciendo uso de *roaming*, está al final del ciclo de facturación del operador de telefonía móvil, o hace uso de un plan de datos limitado de prepago.
98. Al habilitar estas capacidades, las apps compatibles hacen un uso más eficiente y limitado de la conexión de datos móvil (2/3/4G) mientras ejecutan en primer plano, y el sistema limita el uso de datos cuando ejecutan de fondo.
99. El usuario puede crear una lista blanca de apps autorizadas, para las que sí se permitirá hacer uso de las comunicaciones de datos móviles cuando estén ejecutando de fondo incluso aunque Data Saver esté activo. Mediante el menú "Ajustes - Uso de datos - Ahorro de datos - Acceso a datos no restringidos".
100. Desde el punto de vista de seguridad, en Android 7.x caben destacar las mejoras durante el proceso de actualización de Android (o "Seamless Updates"; consultar apartado "6.5. A/B (Seamless) System updates"), de aplicación para los nuevos modelos de dispositivo móvil, llevándose a cabo su ejecución transparentemente (de fondo o en *background*), sin que el usuario tenga que esperar a que se completen las mismas. En el caso de modelos de dispositivos móviles previos, se ha reducido el tiempo necesario para llevar a cabo el proceso de actualización de Android.
101. Asimismo, Android 7.x introduce un nuevo sistema de cifrado del dispositivo móvil basado en ficheros (en lugar de en la partición completa de usuario o de datos), denominado "File-based Encryption" (ver apartado "10.1. Cifrado del dispositivo móvil") y, asociado a éste, un nuevo sistema de arranque

denominado Direct Boot²², que permite un arranque más rápido del dispositivo móvil y que no requiere que el usuario deba introducir el código de acceso durante el proceso de arranque del dispositivo para poder descifrar la partición de usuario, disponiendo así de acceso a funcionalidad básica del dispositivo móvil ante reinicios inesperados (ver apartado "10.1. Cifrado del dispositivo móvil").

102. A raíz de la aparición de la familia de vulnerabilidades conocidas como "Stagefright" [Ref.- 716], se incorporaron a Android 7.x, Nougat, una serie de medidas destinadas a proteger al sistema de futuras amenazas: por ejemplo, se implementaron técnicas de detección de desbordamiento de pila, y el proceso MediaServer se desdobló en varios procesos independientes que ejecutan en su propio sandbox con permisos limitados a sus requerimientos. Así, sólo el proceso AudioServer tendrá acceso Bluetooth, mientras que el proceso MediaCodecService sólo tiene concedido acceso a la tarjeta gráfica o GPU.
103. Android 7.x introduce mecanismos para minimizar la inyección de código en el kernel de Linux, dividiendo la memoria del kernel en segmentos lógicos para código y datos, e impidiendo que el kernel pueda acceder de forma directa al espacio de memoria del usuario.
104. Para los dispositivos que traen de serie Android 7.x se imponen las restricciones introducidas por la característica "Verified Boot", de modo que, si los archivos de sistema se corrompen o modifican con respecto al contenido original y legítimo de la partición de sistema, el sistema operativo puede incluso impedir por completo el arranque.
105. Android 7.x introduce cambios en la gestión de autoridades certificadoras (CAs, *Certificate Authorities*) dirigidos a mejorar la seguridad, tanto del tráfico de las apps como de sus datos [Ref.- 711]. Para ello, las apps basadas en la API 24 y superiores no confiarán por defecto en las CAs añadidas por el usuario (salvo que lo soliciten explícitamente en su sección "<network-security-config>", NSC); además, las CAs ofrecidas por defecto en Android 7.x serán las mismas en todos los dispositivos móviles que ejecuten esta versión, a diferencia de lo que sucedía en versiones anteriores, en las que las CAs podían ser distintas según el modelo y fabricante del dispositivo.
106. Android 7.x incorpora la característica denominada "Network Security Configuration" (NSC) [Ref.- 747], que permite a las apps personalizar sus ajustes de seguridad relacionados con el tráfico de red en un fichero de configuración único por app, sin requerir modificar su código fuente. Estos ajustes se pueden configurar para dominios específicos y para apps específicas, proporcionando, entre otros:
 - *Trust anchors* personalizados: se puede definir qué CAs se consideran de confianza para establecer conexiones seguras mediante HTTPS. Esto permite limitar las conexiones empleando certificados autofirmados o a un subconjunto de CAs públicas, de confianza para la app.

²² <https://android-developers.blogspot.com.es/2016/04/developing-for-direct-boot.html>

- Protección ante el uso accidental de tráfico en texto claro.
 - *Certificate pinning* (CP): restringir el establecimiento de conexiones seguras de la app para únicamente ciertos certificados.
107. Android 7.1 añade soporte nativo para la implementación de teclados de imágenes.
108. Android 7.1 también incorpora APIs que ofrecen soporte para telefonía multi-endpoint (esta característica no está disponible para el usuario final, sino para operadoras y apps de telefonía).
109. Android 7.1.1 introduce las denominadas "app shortcuts", o accesos directos a acciones de las apps, que se muestran al presionar de forma prolongada su icono.
110. Por otro lado, Android 7.x permite personalizar la funcionalidad existente en el menú de accesos o ajustes de configuración rápidos (*custom "Quick Settings"*), disponible en la parte superior de la pantalla de bloqueo del dispositivo móvil (o también desde el menú superior cuando éste está desbloqueado), pudiendo el usuario definir qué accesos y controles están disponibles en todo momento. Esta flexibilidad también se extiende para poder disponer de múltiples páginas con ajustes de configuración rápidos, determinar su ubicación en el panel, e incluso añadir nuevos ajustes proporcionados por las apps.
111. Estas capacidades de personalización tienen un impacto directo en la seguridad del dispositivo móvil y en la funcionalidad disponible en la pantalla de bloqueo, lo que conlleva mejoras notables desde el punto de vista de seguridad cuando un tercero dispone de acceso físico no autorizado al dispositivo móvil (limitando las acciones que puede realizar con el dispositivo móvil sin conocer el código de acceso) [Ref.- 702].
112. Android 7.x complementa el soporte previo para el bloqueo de números de teléfono no deseados, y proporciona una API para que los proveedores de servicio mantengan una lista negra de números bloqueados. Las aplicaciones por defecto de SMS, teléfono y "operador" pueden leer y escribir en esa lista, la cual no es accesible para otras apps, y la misma posibilita que el usuario no reciba comunicaciones innecesarias a través de otros servicios como VoIP o redirecciones²³.
113. Android 7.x incorpora soporte multilenguaje, permitiendo a los usuarios seleccionar más de un lenguaje en los ajustes de configuración, de forma que las apps, a través de una nueva API, pueden consultar los lenguajes elegidos por el usuario y mejorar su experiencia (por ejemplo, mostrando los resultados de una búsqueda en múltiples lenguajes, ofreciendo contenido específico en función de los lenguajes definidos, o no preguntando si se desea traducir páginas a un lenguaje conocido por el usuario). Además, Android 7.x ha añadido soporte para nuevos lenguajes.

²³ <https://developer.android.com/reference/android/provider/BlockedNumberContract.html>

114. Android 7.x incorpora mejoras de accesibilidad, permitiendo al usuario definir el tamaño de pantalla a fin de ampliar o reducir el tamaño de los elementos a modo de zoom. Si el usuario modifica el tamaño de pantalla, Android 7.x informará a todos los procesos (tanto los que están en primer como en segundo plano) del cambio en la configuración. En versiones anteriores de Android, el sistema, ante una modificación en el tamaño de pantalla, mataba los procesos que ejecutaban en segundo plano.
115. Adicionalmente, la pantalla de bienvenida tras la instalación del dispositivo móvil incluye ajustes de visión, que permiten configurar el tamaño de pantalla, el gesto a emplear para magnificar la pantalla, el tamaño de fuente y el mecanismo de TalkBack (ver apartado "9.21. TalkBack").
116. A partir de la versión 51 del navegador web existente por defecto en Android 7.x, Chrome, el APK de Chrome proporciona las Android System WebViews (ver apartado "7.11. Android WebViews"), un componente de navegación web interno, instalado por defecto en Android, que permite a las aplicaciones de Android mostrar contenidos web. Este componente ya estaba preinstalado en los dispositivos Android desde la versión 5.0 a través de un APK WebView, y se actualizaba de forma independiente a través de Google Play para aplicar las últimas actualizaciones de seguridad y otras correcciones de errores. A partir de Android 7.x y Chrome 51, es el APK de Chrome, es decir, el propio Chrome, quien se encargará de mostrar (o renderizar) los contenidos web, proporcionando su implementación para las Android System WebViews a las apps que hagan uso de WebViews. Adicionalmente, WebView ejecutará el contenido web dentro de un proceso contenedor (*sandbox*) independiente, siempre que la opción de desarrollador "Multiprocess WebView" se encuentre habilitada.
117. El contexto JavaScript de las aplicaciones desarrolladas para Android 7.x se reseteará cuando se cargue una nueva página web, a diferencia de versiones previas de Android, donde el contexto inicial se transfiere a las nuevas instancias de WebView.
118. En las aplicaciones diseñadas para Android 7.x, la API de geolocalización sólo estará accesible mediante HTTPS, a fin de proteger la privacidad del usuario cuando se use una conexión no segura.
119. Se introducen nuevas características y APIs para la funcionalidad de Android for Work (ver apartado "8.2. Android for Work"), como el "security challenge" para apps que corren en el perfil de trabajo, presentándose un "work challenge" cuando el usuario trata de abrir alguna aplicación de trabajo (de tipo *work*); también se incorpora la opción "Turn off work" que permite deshabilitar las apps del perfil de trabajo, la sincronización de fondo (*background*) y las notificaciones.
120. Android 7.x elimina el soporte de operaciones criptográficas basadas en el algoritmo SHA1PRNG, recomendándose que los desarrolladores de apps no deriven claves mediante este mecanismo empleando SecureRandom (utilizando una contraseña como semilla) y que en su lugar hagan uso de una función de derivación de claves (o KDF, *Key Derivation Function*) [Ref.- 709].

- Este método no es robusto desde un punto de vista criptográfico y el generador de números aleatorios está influenciado por la semilla empleada.
121. Adicionalmente, Android 7.x también elimina el soporte completo del proveedor criptográfico Crypto.
 122. Android 7.x deshabilita la "cipher suite" (conjunto de algoritmos criptográficos) de tipo RC4 en la configuración TLS/SSL empleada por defecto por las apps para HTTPS, y habilita el CHACHA20-POLY1305.
 123. Android 7.x incrementa la aleatoriedad en ASLR (Address Space Layout Randomization, que es una técnica de protección ante ataques de corrupción de memoria) y en el orden de carga de las librerías, haciendo que los ataques de reutilización de código sean menos fiables [Ref.- 706].
 124. Asimismo, Android 7.x hace uso de la versión 2 del esquema de firma y verificación de los ficheros APK, basado en la firma del contenido binario del fichero APK completo, mejorando la velocidad del proceso de verificación, así como garantizando una mayor integridad de los paquetes (APKs) de las apps de Android [Ref.- 712].
 125. Se impone la política "StrictMode" [Ref.- 724] para la compartición de ficheros entre apps, de forma que ya no sea posible que una app permita acceso a su directorio a otras apps mediante asignación de permisos de lectura y/o escritura globales de Linux.
 126. El directorio privado de las apps ("/data/data/") hace uso de los permisos 0700 de Linux para las apps cuya versión objetivo de Android es el nivel de API 24 (Android 7.x), o superior.
 127. Con el objetivo de aumentar la privacidad de Android, se ha restringido en Android 7.x el acceso por parte de las apps a identificadores persistentes del dispositivo móvil, como por ejemplo las direcciones MAC (del interfaz Wi-Fi o Bluetooth).
 128. A fin de evitar que las apps aborten tras una actualización de la plataforma, o que, para impedirlo, los desarrolladores de las mismas deban realizar actualizaciones de emergencia sobre ellas, Android 7.x restringe el acceso a las librerías nativas privadas con las que el código C/C++ de una app puede enlazar en tiempo de ejecución [Ref.- 710]. Sí se mantiene el enlace con librerías NDK públicas y con las librerías propias de la app.
 129. En Android 7.1 el cifrado basado en FBE (File-based Encryption, ver apartado "10. Cifrado de datos en Android") está habilitado por defecto en algunos dispositivos móviles (a elección del fabricante), en lugar de FDE, y consolida la disponibilidad desde Android 7.x de múltiples particiones duplicadas (boot, system, vendor), habiéndose eliminado la partición de recuperación (recovery) y de cache²⁴.
 130. El directorio raíz ("/") está asociado a "initramfs" en Android 7.x con "seamless updates" y pertenece a la partición de sistema, en lugar de a la

²⁴ <https://plus.google.com/+Chainfire/posts/fvEPo42GKXS>

partición de arranque (boot, tal y como ocurría en versiones previas de Android). La funcionalidad de recuperación (recovery) se encuentra integrada dentro de la partición de arranque (boot).

131. El lanzamiento de Android 7.x, junto a sus numerosas novedades de seguridad, fue acompañado posteriormente por la publicación de un nuevo portal de seguridad asociado a Android por parte de Google, el *Android Security Center*, en noviembre de 2016 [Ref.- 713].
132. Dentro de este portal se ofrece información sobre las nuevas características de seguridad de Android, el proceso asociado a los boletines y actualizaciones de seguridad, o las publicaciones oficiales de Google sobre la seguridad de Android, por ejemplo:
 - La definición de apps potencialmente dañinas o PHAs (Potentially Harmful Applications) [Ref.- 714], que incluye una clasificación en función del tipo de debilidad al que se asocia la app (por ejemplo, denegación de servicio, backdoor, escalada de privilegios en el sistema, ransomware...). Esta clasificación se usará por el mecanismo de Verify Apps para alertar al usuario ante la instalación de una app potencialmente dañina (ver apartado "Verificación de seguridad de las apps: Google Play Protect" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]).
 - Los informes anuales de revisión de la seguridad global en Android [Ref.- 24].
 - Las técnicas empleadas para mantener las apps potencialmente dañinas fuera de Google Play, entre las que se incluyen análisis estáticos y dinámicos del código, informes realizados por terceros pertenecientes al ámbito académico y empresarial, el servicio SafetyNet (detallado en el apartado "SafetyNet" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]).

4.11. DIFERENCIAS ENTRE LAS MÚLTIPLES VERSIONES DE ANDROID

133. Las principales diferencias y novedades para los usuarios de cada una de las versiones de Android están disponibles en la página web oficial de Android, incluyendo la información histórica para las diferentes versiones [Ref.- 202]: 2.2, 2.3, 3.0, 4.0 (Ice Cream Sandwich, ICS), 4.1, 4.2 y 4.3 (Jelly Bean, JB), 4.4 (KitKat, KK), 5.0 (Lollipop, L), 6.0 (Marshmallow, M) y 7.0 (Nougat, N). En la misma destacan las novedades de las versiones más recientes:
 - 4.3 (Jelly Bean): <https://www.android.com/versions/jelly-bean-4-3/>
 - 4.4 (KitKat): <https://www.android.com/versions/kit-kat-4-4/> [Ref.- 148]
 - 5.0 (Lollipop): <https://www.android.com/versions/lollipop-5-0/> [Ref.- 500]
 - 6.0 (Marshmallow): <https://www.android.com/versions/marshmallow-6-0/> [Ref.- 600]
 - 7.0 (Nougat): <https://www.android.com/versions/nougat-7-0/> [Ref.- 700]

134. Las principales diferencias y novedades técnicas de cada una de las versiones de la plataforma Android están disponibles en la página web para desarrolladores de Android (Developers) [Ref.- 86]. La documentación para las versiones más recientes, con subapartados para las subversiones principales (por ejemplo, Android 7.1), junto a los detalles para los desarrolladores de las APIs o librerías de programación, los cambios más relevantes, y ejemplos de código, están disponibles en:

- 4.4 (KitKat): [Ref.- 86]
 - <https://developer.android.com/about/versions/kitkat.html>
- 5.0 (Lollipop): [Ref.- 501]
 - <https://developer.android.com/about/versions/lollipop.html>
- 6.0 (Marshmallow): [Ref.- 601]
 - <https://developer.android.com/about/versions/marshmallow/index.html>
- 7.0 (Nougat): [Ref.- 701]
 - <https://developer.android.com/about/versions/nougat/index.html>

135. Las principales características de seguridad y mecanismos de protección existentes en Android, especialmente para las últimas versiones, están disponibles en la página oficial de seguridad de Android para los usuarios finales [Ref.- 132], y en la página oficial de seguridad de Android para usuarios técnicos [Ref.- 133]. Asimismo, las mejoras de seguridad introducidas por cada una de las versiones de Android están resumidas en la página oficial de mejoras de seguridad de Android, también clasificadas por versiones [Ref.- 140].

136. Por otro lado, las principales características y mejores prácticas asociadas a la privacidad y seguridad de la plataforma Android están disponibles para los desarrolladores de apps en una página web dedicada [Ref.- 57], entre las que se encuentran el uso de HTTPS y TLS, el uso de configuraciones de seguridad de red, la actualización del proveedor de seguridad para evitar la explotación de vulnerabilidades de SSL/TLS, la integración con SafetyNet, la comprobación de URLs mediante la API de navegación segura (*Safe Browsing*), la gestión de claves mediante una Keystore hardware, la mejora de la seguridad mediante el uso de políticas de gestión de los dispositivos móviles, o soporte para arranque directo, entre otros (todas ellas descritas en los apartados correspondientes de la presente guía).

137. En resumen, los recursos de Android oficiales principales se engloban en:

- Las características de Android para usuarios finales [Ref.- 203]:
 - 4.4 ([Ref.- 148]), 5.x ([Ref.- 500]) y 6.x ([Ref.- 600]) y 7.x ([Ref.- 700]).
- Las características de seguridad de Android para usuarios finales [Ref.- 132].
- Las características de seguridad de Android para usuarios técnicos [Ref.- 133].

- Las mejoras de seguridad de Android para cada una de las versiones [Ref.- 140].
- Las características técnicas de Android para desarrolladores para cada una de las versiones [Ref.- 86].
- Las mejores prácticas de privacidad y seguridad de Android para desarrolladores [Ref.- 57].

5. ENTORNO DE APLICACIÓN DE ESTA GUÍA

138. Los contenidos de esta guía aplican a dispositivos móviles basados en Android en castellano: "Español (España)". El idioma puede ser seleccionado desde el menú "Ajustes [Personal] - Idioma e introducción de texto - [Preferencias de idioma]" y pulsando sobre el idioma de preferencia a la vez que se desplaza el dedo hacia la parte superior.

Nota: El menú "Ajustes" empleado y referenciado en numerosas tareas de configuración del dispositivo móvil Android está disponible desde el menú de ajustes de configuración rápidos, o ajustes rápidos por abreviar (situado en la barra superior de estado) del dispositivo móvil, o desde el icono del *Launcher* (icono con una matriz de puntos situado en la parte inferior central de la pantalla principal o *Home* de Android).

139. La guía ha sido probada y verificada con la versión 7.1.2 del sistema operativo Android, si bien algunos apartados pueden corresponder con otras versiones 7.x intermedias, como la 7.0 o la 7.1.1, especialmente los relacionados con la actualización del sistema operativo o la instalación inicial del dispositivo móvil.

Nota: La información general del dispositivo móvil está disponible bajo la opción "Ajustes [Sistema] - Información del teléfono", y en concreto, la información de la versión del sistema operativo se encuentra bajo la sección "Versión de Android".

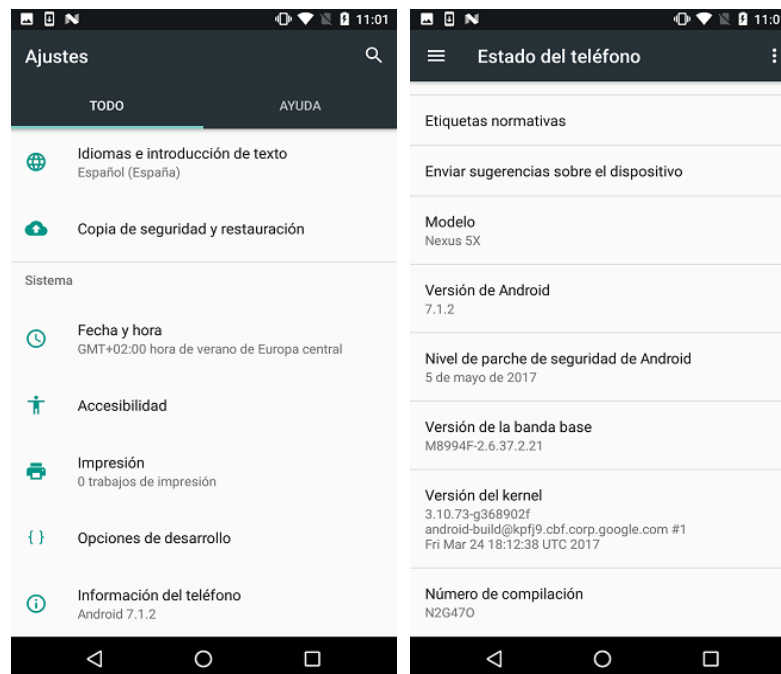
140. Las operaciones que requieren del uso de un ordenador han sido llevadas a cabo mediante Windows 10 Home Premium 64-bits en castellano.

Nota: La presente guía emplea el término "ordenador" para referenciar al equipo portátil o de sobremesa (o escritorio) empleado para la sincronización, gestión y tareas de administración, depuración y desarrollo sobre el dispositivo móvil.

141. El hardware sobre el que se ha verificado la presente guía tiene las siguientes características:

- Dispositivo móvil Google Nexus 5X (fabricado por LG, modelo LG-H791) [Ref.- 3]:
 - Versión de Android: 7.1.2. Es importante reseñar que el dispositivo móvil dispone de fábrica de la versión Android 7.0 precargada, por lo que algunos aspectos de configuración tras la instalación inicial (incluida la actualización a la versión 7.1.2) pueden estar basados en la versión 7.0.

- Versión de banda base: (ver imagen inferior): M8994F-2.6.37.2.21
- Versión de kernel: (ver imagen inferior): 3.10.73-gd79df7e45e ... Junio ... 2017
- Número de compilación: MTC20K



Nota: El dispositivo móvil empleado es un terminal libre, es decir, no asociado a ningún operador de telefonía móvil, con el objetivo de analizar las características y valores por defecto de Android sin verse afectadas por las posibles modificaciones y personalizaciones llevadas a cabo por los operadores de telefonía.

Nota: La guía ha sido diseñada (y aplicada) para dispositivos móviles estándar basados en Android en los que no se ha llevado a cabo el proceso de *rooting* o root. Dicho proceso permite al usuario disponer de control completo del dispositivo móvil y acceder al mismo como root, superusuario, o usuario privilegiado, y en concreto, al subsistema Linux subyacente en Android, eliminando así los controles y/o restricciones establecidos por la plataforma Android estándar, los fabricantes y/o los operadores de telefonía móvil asociados al dispositivo móvil.

Las apps disponibles para los dispositivos móviles Android, por defecto, no disponen de permisos de escritura en ciertas zonas del sistema de ficheros, no pudiendo reemplazar el sistema operativo, ni de privilegios para hacer uso de capacidades específicas, como reiniciar el dispositivo móvil, modificar el uso de los LEDs hardware, capturar la pantalla del dispositivo móvil o eliminar aplicaciones móviles instaladas por algunos operadores de telefonía móvil. El proceso de *rooting* es utilizado por usuarios avanzados para acceder a estas funcionalidades de bajo nivel.

Los dispositivos móviles Android *rooteados* (o *rooted*) ignoran el modelo de seguridad descrito a lo largo de la presente guía, ya que todas las aplicaciones que sean instaladas podrán disponer de los máximos privilegios en el dispositivo (root), exponiendo potencialmente a los usuarios a código dañino que podría tomar control completo del terminal. Android no requiere llevar a cabo el proceso de *rooting* para permitir la instalación de apps de terceros no oficiales, es decir, no distribuidas a través de Google Play, tal como sucede en otras plataformas de dispositivos móviles, como iOS, donde es necesario realizar el proceso de *jailbreak* (equivalente a *rooting*) para poder instalar ese tipo de apps no oficiales.

Las particularidades del proceso de actualización en Android (ver apartado "0.

Actualización del sistema operativo Android"), con largos periodos de tiempo hasta que una nueva actualización está disponible para algunos modelos y usuarios, ha sido a lo largo de la historia uno de los motivos que incentiva a muchos usuarios a realizar el proceso de *rooting* para así poder instalar la última versión de Android en su dispositivo móvil empleando otros proyectos como LineageOS (CyanogenMod).

Nota: Debe tenerse en cuenta que se pueden presentar diferencias en la apariencia de las capturas de pantalla utilizadas a lo largo de la presente guía y la pantalla de otros dispositivos móviles basados en la misma versión de Android, debido al modelo, fabricante o versión concreta del SO.

Asimismo, algunas de las funcionalidades disponibles en Android pudieran ser propias de un fabricante de dispositivos móviles determinado, por lo que las recomendaciones de seguridad asociadas no aplicarían a otros dispositivos móviles basados en Android. Se recomienda llevar a cabo un análisis de seguridad similar al mostrado para estas funcionalidades adicionales añadidas por los fabricantes con el objetivo de determinar las recomendaciones a aplicar en dispositivos móviles de otros fabricantes y en sus aplicaciones propietarias.

La versión 7.1.2 de Android para el dispositivo móvil empleado para la elaboración de la presente guía (Nexus 5X) está disponible como actualización parcial OTA (Over-the-Air) partiendo de la versión 7.0 ó 7.1.1 de Android, y complementariamente a la imagen de fábrica completa de esta versión de sistema operativo²⁵, desde diciembre de 2015 (ver apartado "0.

Actualización del sistema operativo Android"). La actualización manual aplicando una imagen OTA completa de una versión de Android no requiere volver a realizar el proceso de configuración inicial del dispositivo móvil, ni elimina información alguna del dispositivo.

142. Se recomienda siempre recabar e inventariar la información de versiones, tanto hardware como software, de los dispositivos móviles a proteger, con el objetivo de disponer de toda la información necesaria a la hora de tomar decisiones relativas a la aplicación de nuevas actualizaciones y medidas de seguridad.

²⁵ <https://developers.google.com/android/nexus/images>

143. Debe prestarse atención al contrato de telefonía, y en concreto de datos móviles, asociado a la tarjeta SIM (*Subscriber Identity Module*) empleada en el dispositivo móvil Android. La ausencia de capacidades de comunicación de datos es notificada al usuario en función de las apps que requieren disponer de las mismas.
144. Igualmente, debe prestarse especial atención al tipo de dispositivo móvil (marca y modelo) al que se desea aplicar la guía, ya que la relación de los elementos descritos o referenciados en ella (interfaz de usuario, capacidades hardware, aplicaciones, etc.) puede diferir ligeramente en función de la personalización del sistema operativo realizada por el fabricante del terminal (u operador de telefonía móvil).
145. Antes de aplicar esta guía en un entorno de producción debe confirmarse su adecuación a la organización objetivo, siendo probada previamente en un entorno aislado y controlado donde se puedan realizar las pruebas necesarias y aplicar cambios en la configuración para que los mismos se ajusten a los criterios y requisitos específicos de cada organización.

Nota: El alcance de la presente guía no contempla la realización de un análisis de seguridad detallado de los protocolos de comunicaciones basados en HTTPS (o SSL/TLS) y empleados por los dispositivos móviles Android, las aplicaciones móviles disponibles por defecto (o de terceros) y los diferentes servicios de Google (o de terceros). Se recomienda consultar la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407] para obtener detalles sobre los servicios asociados a una cuenta de Google.

6. ACTUALIZACIÓN DEL SISTEMA OPERATIVO ANDROID

146. Para llevar a cabo la actualización de un dispositivo móvil a una nueva versión de Android, existen dos alternativas: llevar a cabo una actualización manual o dejar que el sistema operativo realice una actualización automática.
147. En el primer caso, es el propio usuario el que, activamente, debe identificar la disponibilidad de una nueva actualización de la versión de Android (considerando entre ellas las actualizaciones de seguridad que se suministran mensualmente por parte de Google).
148. En el segundo caso, es el propio sistema operativo el que, en base a su configuración y a los mecanismos internos del sistema, identificará y aplicará las actualizaciones, bien de forma automática, bien con cierta interacción por parte del usuario.
149. Ambas opciones (manual y automática) se detallan en los siguientes apartados.
150. Es importante recalcar que las actualizaciones (tanto automáticas como manuales) pueden ir encaminadas a instalar en el dispositivo móvil:

- Una nueva versión "major" (por ejemplo, actualización de la versión 6.0.1 de Android a la versión 7.0).
- Una nueva versión "minor" (por ejemplo, actualización de la versión 7.1.1 de Android a la versión 7.1.2).
- Un *bundle* de parches de seguridad sobre una versión concreta (por ejemplo, la aplicación de los parches de seguridad para Android 7.1.2 de septiembre de 2017 sobre un dispositivo que ejecuta Android 7.1.2 con el nivel de parches de mayo de 2017).

151. A su vez, las actualizaciones pueden ser:

- Parciales: el paquete de instalación no contiene todos los paquetes y contenidos propios de una versión estable completa del sistema operativo, sino sólo los cambios que se deben aplicar sobre una versión concreta para pasar a la versión objetivo. Por este motivo, los paquetes de actualización parcial no sirven para devolver un dispositivo a una versión completa después de un fallo crítico.
- Completas: el paquete de instalación contiene todos los paquetes y contenidos propios de la versión de Android a la que pertenecen, pudiendo instalarse directamente en el dispositivo móvil siguiendo los procedimientos adecuados para disponer de la versión objetivo.

152. Google no libera públicamente las actualizaciones parciales para su instalación manual, este proceso se lleva a cabo a través de paquetes OTA que el dispositivo descarga de los servidores de Google (aunque es posible identificar dónde están ubicadas dichas actualizaciones parciales mediante el análisis del tráfico de red).

153. Respecto a las imágenes o actualizaciones completas, Google mantiene públicamente dos tipos de imagen:

- Imágenes de fábrica (*Factory images*): contienen todos los ficheros binarios para dejar el dispositivo móvil en la versión asociada a la imagen en cuestión (al aplicarlas, el dispositivo pierde su configuración y queda tal como marca la versión de fábrica).
- Imágenes OTA completas (*Full OTA images*): contienen todos los paquetes de actualización OTA para una versión concreta compatible con la existente en el dispositivo móvil. Para instalarlas no se requiere borrar por completo el dispositivo, es decir, se mantiene la configuración actual.

6.1. ACTUALIZACIÓN AUTOMÁTICA DEL SISTEMA OPERATIVO ANDROID

154. Los dispositivos móviles Android incorporan mecanismos que permiten actualizar el sistema operativo de forma sencilla para el usuario.

155. Los dispositivos móviles Android, una vez disponen de conectividad de datos hacia Internet a través de las redes de telefonía móvil (2/3/4G) o de redes Wi-Fi, pueden verificar automáticamente si existe una nueva versión o actualización del sistema operativo.

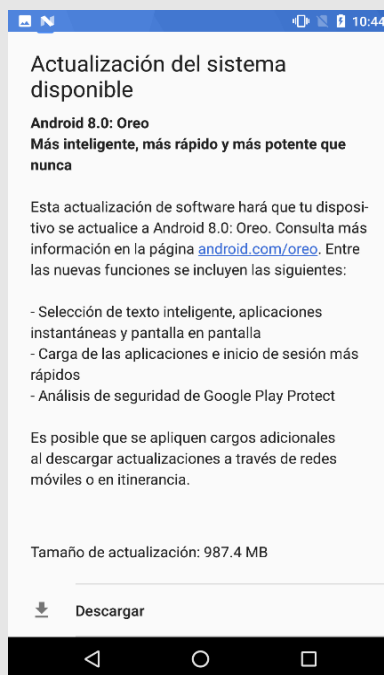
156. Para ello, se establecen conexiones tanto cifradas mediante HTTPS (TCP/443) con los servidores "http[s]://android.clients.google.com" y "https://www.googleapis.com".
157. Para realizar las comprobaciones de versión llevadas a cabo por Android se utiliza una petición al recurso "/checkin", de tipo POST, que emplea como parámetros datos en formato ProtoBuffer ("application/x-protobuf"; comprimidos mediante GZIP²⁶), es decir, empleando el formato binario de intercambio de datos utilizado por muchos servicios de Google y conocido como Google Protocol Buffers²⁷. Como respuesta, los servidores de Google envían también datos en formato ProtoBuffer, entre los que se incluye la versión más reciente de Android que se puede descargar a partir de la versión actual del terminal.
158. Las actualizaciones automáticas se basan en el concepto de actualizaciones OTA, "OTA update" (Over-The-Air update), definido como el proceso de servir, a través de una conexión inalámbrica, una actualización de software a un dispositivo móvil capacitado para recibirla sin procedimientos complejos. El concepto OTA no es propietario de Google, sino que se aplica de forma genérica a los mecanismos de intercambio de información entre dispositivos móviles y sus proveedores (de software, firmware, operador de telecomunicación, etc.) destinados a actualizar sus componentes.
159. Un proceso de actualización se inicia cuando el paquete OTA se libera públicamente para el dispositivo móvil concreto, y las políticas del dispositivo permiten su procesamiento (por ejemplo, el estado de la batería o la actividad que en ese momento esté llevando a cabo el usuario, pueden diferir la descarga e instalación de la actualización. El fichero de actualización contiene metadatos (incluyendo una lista de acciones para verificar que la nueva versión es de aplicación en el dispositivo objetivo), y los propios contenidos y binarios que forman parte de la actualización del sistema.
160. En el caso de Android 7.0, la versión que se ofrecía como actualización OTA para el Nexus 5X a fecha de redacción de la presente guía era la 7.1.1, a pesar de que la versión 7.1.2 ya se encontraba disponible.
161. Si se desea actualizar un dispositivo a una versión de Android 7.x diferente a la que Google ofrece para el dispositivo en un momento concreto, se puede descargar la imagen OTA completa de la versión deseada de la web oficial de Google y realizar una actualización manual completa (ver apartado "6.3. Actualización manual completa de Android"). Alternativamente, se podría descargar la imagen de fábrica completa correspondiente a la versión deseada (por ejemplo, la 7.1.2) y llevar a cabo el proceso de "flashing" descrito en el mismo apartado.

Nota: Se ha constatado que, a fecha de publicación de la presente guía, la versión de Android 8.0 (Oreo) se ofrecerá como actualización para los terminales Nexus 5X con

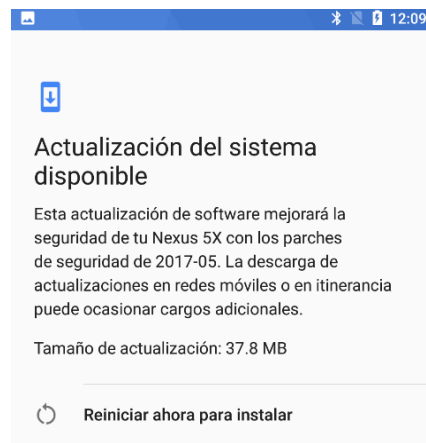
²⁶ Dependiendo de la versión de las apps Google Services Framework (GSF) y Google Play Services (GPS).

²⁷ <https://github.com/google/protobuf/>

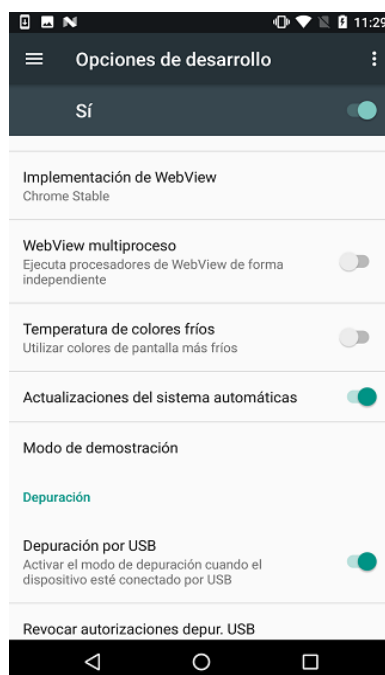
versión 7.x de Android. Sin embargo, no se detalla este proceso de actualización por quedar la versión 8.0 de Android fuera del alcance de la presente guía.



162. En Android 7.x, se ha reducido el tiempo necesario de instalación de las actualizaciones OTA, y los ficheros de actualización son de un tamaño reducido. No es necesario esperar a que se lleve a cabo la optimización de las apps (la parte más lenta del proceso de actualización), sino que el nuevo compilador JIT (Just in Time) ha sido optimizado para hacer que el proceso de instalación y actualización sea mucho más rápido [Ref.- 706].
163. Android 7.x introdujo el mecanismo de actualización denominado "Seamless updates" (ver apartado "6.5. A/B (Seamless) System updates"), que redundaba en una clara mejora en el rendimiento del proceso de instalación.
164. En el caso de Android 7.x, nada más completarse la instalación inicial, el dispositivo se conecta a <https://android.clients.google.com> y comienza a descargarse el parche de seguridad de marzo de 2017, sin solicitar ningún tipo de confirmación por parte del usuario.
165. Si, tras actualizarse al parche de marzo de 2017 se hiciese un reseteo a fábrica, la versión que se instalaría sería la 7.1.1 con el parche de marzo.
166. Una vez actualizado el dispositivo al parche de marzo de 2017, el dispositivo volverá a realizar una comprobación y mostrará la disponibilidad de una nueva versión del sistema, pero no se indica concretamente que sea la 7.1.2. Para esta versión, tampoco se solicita confirmación al usuario para iniciar la descarga, la cual se produce automáticamente, y tras ello, se informa al usuario de que reinicie para proceder a instalarla. Por tanto, no es posible optar por mantenerse en la versión 7.1.1 una vez que el dispositivo detecta que puede descargarse la 7.1.2.



167. Este comportamiento de descargas automáticas de actualizaciones es debido a una opción que introdujo Android 7.x dentro de las opciones de desarrollo denominada "Actualizaciones del sistema automáticas" (accesible a través de "Ajustes - [Sistema] Opciones de desarrollo"). Con esta opción (activa por defecto), el dispositivo móvil recibirá de los servidores de Google (mediante un procedimiento de tipo "push") las nuevas versiones de OTA disponibles, y procederá a su descarga e instalación sin solicitar confirmación al usuario siempre que se cumpla la condición de estar conectado a una red Wi-Fi y que cuente al menos con un 50% de batería.



168. Pese a que se recomienda encarecidamente mantener el dispositivo móvil Android en la versión más reciente (tanto a nivel de versión consolidada como de nivel de parches de seguridad publicados para una versión de Android concreta), en determinados escenarios puede resultar conveniente desactivar la opción "Actualizaciones del sistema automáticas" y proceder a la instalación de la última versión de parches de seguridad tras haber analizado su contenido y si alguna opción que se haya incorporado puede provocar comportamientos no deseados en el dispositivo.

169. Es necesario destacar que esta opción solo está activa por defecto para los dispositivos móviles que disponen de Android 7.0 o superior precargado de fábrica, y no para aquellos que, estando en una versión anterior de Android, hayan sido actualizados a Android 7.x.
170. Es también importante añadir que esta opción no aplica a actualizaciones hacia versiones de Android superiores. En el caso de Android 7.x, no se llevará a cabo la actualización automática a Android 8.x, aunque la opción "Actualizaciones del sistema automáticas" se encuentre activa.
171. Una cuestión muy importante es que, una vez se detecta que existe una versión mayor de Android disponible, los parches de seguridad correspondientes a la versión actual no se aplicarán, siendo necesario instalarlos manualmente siguiendo el procedimiento descrito en el apartado "6.4. Actualización manual parcial de Android". El motivo de ello es que la política de actualizaciones de Google pretende que sus terminales ejecuten la última versión de Android principal que haya sido liberada para ellos.
172. El agente de usuario (User-Agent) empleado para la comunicación HTTP/HTTPS en Android 7.1.1 es:

```
User-Agent: Dalvik/2.1.0 (Linux; U; Android 7.1.1; Nexus 5X  
Build/N4F26O
```

173. Como respuesta a la conexión desde el dispositivo móvil, el servidor "android.clients.google.com" envía otro protobuffer proporcionando multitud de parámetros, incluida la versión a la que se actualizará "update_title Android 7.1.2" y un "update_token" que el dispositivo utilizará en el momento de la descarga en la cabecera "Authorization". El agente de usuario empleado es "User-Agent: Dalvik/2.1.0 (Linux; U; Android 7.1.1; Nexus 5X Build/N4F26O)".

174. La descarga efectiva del fichero (esto aplica en ambos casos: actualización con y sin intervención del usuario) se realiza mediante una petición HTTPS al servidor "https://android.googleapis.com", usando un método GET al recurso
"/packages/ota/google_bullhead_bullhead/<valor_SHA1_del_fichero_ZIP_de_la_actualización>.zip", y en el que se pasa en la cabecera "Authorization" el token recibido como respuesta al recurso "/checkin" anterior. Por ejemplo, para la versión 7.1.2 de Android, el valor SHA1 del fichero ZIP es 7d843620ee1ee00f8e8264c87f6065d746948bac, por lo que el recurso concreto que se solicita es:

```
https://android.googleapis.com/packages/ota/google_bullhead_bullhea  
d/7d843620ee1ee00f8e8264c87f6065d746948bac.zip
```

175. Esta opción es empleada frecuentemente para la distribución de actualizaciones de Android basada en mensajes OTA (Over-The-Air), y complementa la opción por parte del usuario de descargar manualmente el fichero correspondiente a la actualización completa directamente de los servidores oficiales de Google (analizada en el apartado "6.3. Actualización manual completa de Android").

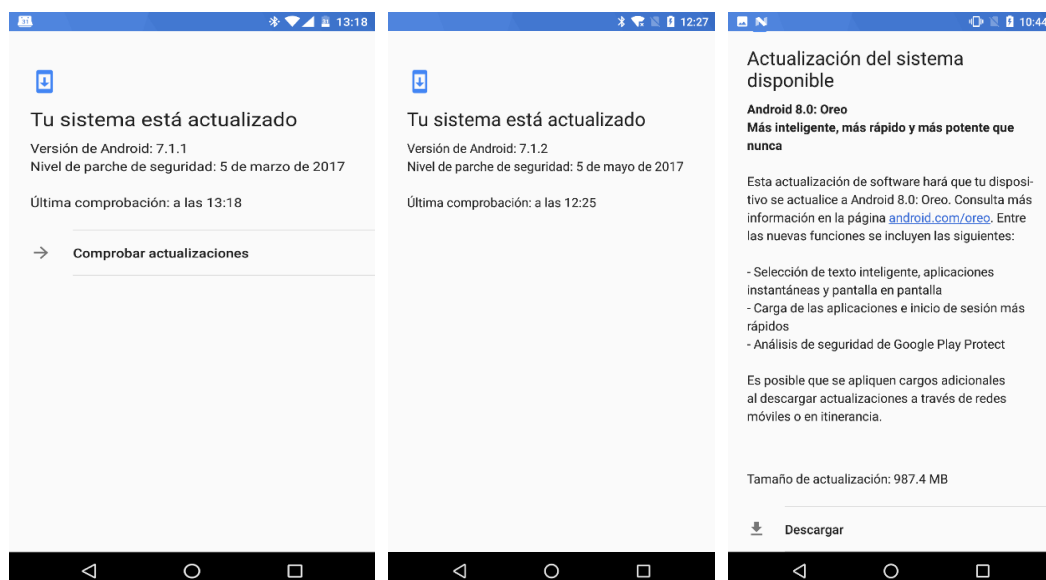
176. Una vez actualizada la versión de Android a la 7.1.2 y reiniciar, el dispositivo volverá a comprobar si existen actualizaciones mediante una petición HTTPS al servidor "https://android.clients.google.com", al recurso "/checkin" mediante el método POST, enviándose como "Content-type" un protobuf. El agente de usuario (User-Agent) empleado para la comunicación es "Dalvik/2.1.0 (Linux; U; Android 7.1.2; Nexus 5X Build/N2G47F)".

177. Como respuesta, el servidor envía otro Protobuffer proporcionando varios parámetros, entre ellos el update_token y el path al fichero de descarga "update_url <enlace>", donde el nombre del fichero ZIP del "<enlace>" vuelve a ser el SHA1 del fichero de actualización:

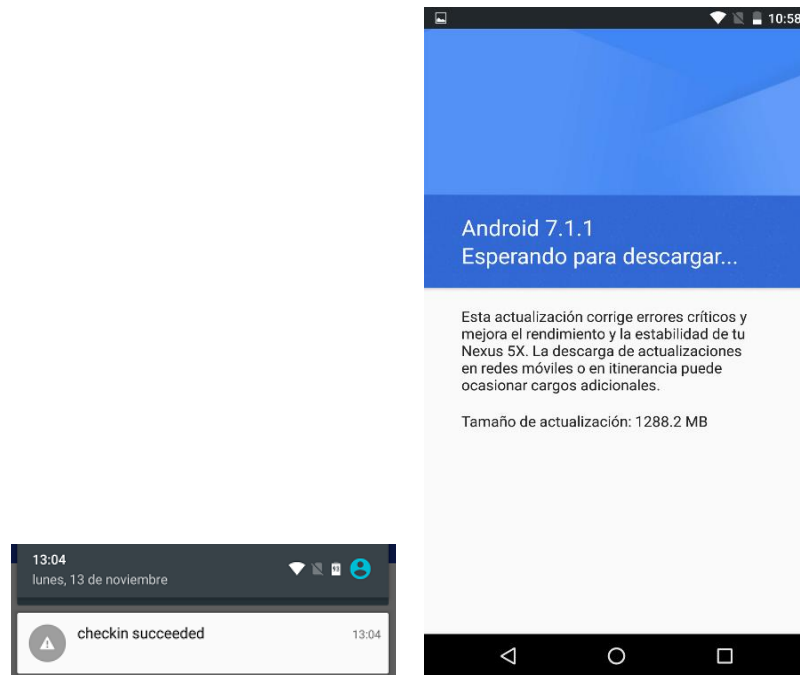
```
https://ota.googlezip.net/packages/ota-api/google_bullhead_
bullhead/a9f4fe0887c33f50db027480550c7b2ec49e231a.zip
```

178. Con la opción "Actualizaciones del sistema automáticas" activa, el dispositivo inicia la descarga sin solicitar confirmación por parte del usuario, mediante una comunicación HTTPS con el servidor "ota.googlezip.net" y pasando en la cabecera "Authorization" el token recibido como parte de la respuesta al recurso "/checkin".

179. En la versión 7.1.1, la pantalla "Ajustes - [Sistema] - Información del teléfono - Actualizaciones del sistema" presentaba la opción "Comprobar actualizaciones". En la versión 7.1.2, esta opción se sustituye por "Actualizaciones del sistema", que informa de cuándo se realizó la última comprobación, o muestra la existencia de una actualización pendiente, en caso de existir una nueva actualización a la que el dispositivo no pueda actualizarse automáticamente, bien por encontrarse deshabilitada la opción "Actualizaciones del sistema automáticas" (accesible a través de "Ajustes - [Sistema] Opciones de desarrollo"), bien porque la actualización disponible corresponda a una versión mayor de Android (ver imagen inferior derecha):



180. Sin embargo, sigue vigente la utilización del código USSD "*##*#checkin##*#" (o "*##*#2432546##*#") desde el teclado del teléfono en Android para forzar una búsqueda inmediata de actualizaciones y parches. Cuando se emplea esta opción, se mostrará una notificación que indica si fue posible (*checkin succeeded*) realizar la comprobación de actualizaciones.



181. En el caso de la actualización desde Android 7.1.1 a Android 7.1.2 en un escenario en el que la opción "Actualizaciones del sistema automáticas" descrita anteriormente se encuentre deshabilitada, la selección del botón "Descargar" hará que Android establezca una conexión cifrada HTTPS con el servidor "https://android.clients.google.com" para descargar la actualización que se encuentre disponible, para lo cual empleará el agente de usuario "AndroidDownloadManager/7.1.1 (Linux; U; Android 7.1.1; Nexus 5X Build/N4F26I)", desvelando la versión de Android y el modelo de dispositivo móvil (ver detalles en el ejemplo inferior; en este caso, se parte de la versión de Android 7.1.1 (N4F26I)). Como respuesta se recibe una redirección (HTTP 302) a uno de los servidores de caché de Google para obtener la nueva actualización:

```
HTTP/1.1 302 Found
X-GUploader-UploadID:
AEnB2Uo8vVcagbBRAFEi5t9s25cmbZ4uyNlVe8oV_TZFPxMChmZfpm9AC2qwuhrQOzCmBw-
2Z1ykLS0HL0PP64YQZjn30J6FfQzH1Sz_YOlzXxtBRYCWfk

Location: https://redirector.gvt1.com/packages/data/ota-api/
google_bullhead_bullhead/d405d41e68f71449cb5d860823962b2e5778330c.zip...
```

182. La integridad de las actualizaciones de las nuevas versiones de Android se verifica a través de una firma digital asociada al fichero de actualización (.zip) descargado, en concreto, mediante los certificados digitales (x.509) de sistema disponibles por defecto en "/system/etc/security/otacerts.zip"²⁸ (si no se especifica un conjunto de certificados distinto), que identifican quién puede distribuir y firmar actualizaciones OTA [Ref.- 44] [Ref.- 160].

183. La ausencia de cifrado al hacer uso de HTTP podría permitir la manipulación de este intercambio de datos, por ejemplo, mediante la sustitución del fichero ZIP asociado a la actualización OTA por otro fichero asociado a otra

²⁸ <http://developer.android.com/reference/android/os/RecoverySystem.html>

versión, también correspondiente a una actualización válida firmada digitalmente.

184. Por otro lado, la utilización de un canal no cifrado HTTP permite a un atacante conocer la versión y compilación actual (y potencialmente futura) del dispositivo móvil víctima, pudiendo explotar vulnerabilidades conocidas sobre dichas versiones.
185. El fichero con la actualización OTA es almacenado en la partición `"/cache"` (partición únicamente accesible como root o superusuario), en concreto, en un fichero `".zip"` cuyo nombre contiene el valor SHA1 del fichero ZIP de la actualización.
186. En base a la documentación y referencias oficiales (incluido el propio mensaje de anuncio inicial), las notificaciones y actualizaciones de seguridad de Android deberían estar disponibles en el grupo o lista de distribución "Android Security Announcements", creada en 2008. Sin embargo, esta lista únicamente contiene una entrada sobre el anuncio inicial del equipo de seguridad de Android de agosto de 2008 [Ref.- 66].
187. Sin embargo, sí permanece activa la lista mencionada en el anuncio inicial (y descrita posteriormente), "Android Security Discussions" [Ref.- 67], no así la lista de preguntas frecuentes de seguridad de Android [Ref.- 68], reemplazada por las mejores prácticas de seguridad.
188. Google publica mensualmente a través del grupo "Android Security Updates" [Ref.- 69] el boletín de seguridad correspondiente a las actualizaciones del nivel de parche de seguridad. El proceso de actualizaciones mensuales lleva asociado una imagen OTA que contiene los parches de seguridad de ese mes para cada versión de Android bajo soporte.
189. Desafortunadamente no se dispone de un repositorio oficial de notificaciones con el listado de vulnerabilidades que han afectado a Android, y/o que han sido solucionadas, y en consecuencia, no existe un listado de actualizaciones disponibles para resolver esas vulnerabilidades (con la excepción descrita posteriormente para los dispositivos móviles Google Nexus y/o Pixel desde agosto del año 2015).
190. Existe otro grupo o lista de distribución para la discusión de temas relacionados con seguridad en Android, denominado "Android Security Discussions" [Ref.- 67]. La problemática actual respecto al modelo de actualizaciones de seguridad de Android (reflejado posteriormente) ha sido analizada y discutida en numerosas ocasiones en dicho grupo.
191. Desde un punto de vista no oficial, existe un proyecto de la comunidad, denominado "Android Vulnerabilities" [Ref.- 64] que actúa como repositorio de las diferentes vulnerabilidades de seguridad que han afectado a las múltiples versiones de Android (al menos hasta que Google decidió oficialmente publicar los boletines de seguridad mensuales). El proyecto recibió contribuciones de la comunidad sobre nuevas vulnerabilidades, y permitía clasificarlas por fabricante, año, versión de Android y contribuyente.

192. Oficialmente, las actualizaciones de Android, al tratarse de un sistema operativo de código abierto basado en Linux y con modificaciones particulares según los modelos de terminal, deben ser proporcionadas por los diferentes fabricantes de los dispositivos móviles [Ref.- 68] u operadores de telefonía móvil, en base a las modificaciones y ajustes realizados sobre las actualizaciones proporcionadas por el proyecto Android. Google, como líder del proyecto Android, no proporciona directamente actualizaciones para los dispositivos móviles Android (salvo para sus propios modelos de la familia Google Nexus y/o Pixel). Concretamente, empleando como ejemplo el fabricante del dispositivo móvil empleado en esta guía, LG dispone de una página web referente a boletines de seguridad para otros modelos de dispositivos móviles [Ref.- 70].

Nota: Debido a que el dispositivo móvil Android empleado para la elaboración de la presente guía es distribuido por Google directamente (actuando como fabricante del mismo, aunque en realidad pueda ser fabricado por otra compañía con la que Google ha llegado a un acuerdo para cada modelo concreto), los procesos de actualización descritos en este documento afectan únicamente a los dispositivos móviles comercializados por Google, dentro de la familia Google Nexus [Ref.- 75], también conocidos como *Google Experience Devices* [Ref.- 72] o *Pure Android, 100% Google*. Los dispositivos Google Nexus están (o han estado) disponibles desde Google Play directamente (también referidos en el pasado como *Google Play Devices* [Ref.- 71]).

Adicionalmente, existieron en el mercado dispositivos móviles conocidos como *Google Play Edition Devices*, es decir, comercializados por Google desde Google Play [Ref.- 71] pero creados por otros fabricantes, fuera de la familia Google Nexus. Estos dispositivos móviles también utilizan una versión no personalizada de Android, y por tanto, las actualizaciones para los mismos eran parejas en el tiempo a las proporcionadas por Google para la familia Google Nexus. Ejemplos de dispositivos móviles de este tipo han sido las ediciones especiales para Google Play del Samsung Galaxy S4, del HTC One o del Motorola Moto G, referenciados normalmente por su nombre seguidos del calificativo "edición Google Play". El programa de los *Google Play Edition Devices* parece haber finalizado en enero de 2015, con la retirada del último modelo disponible, el HTC One M8²⁹.

193. No existe ningún estándar ni modelo definido para la distribución de las actualizaciones de seguridad de Android, ni ninguna periodicidad definida para su publicación (salvo para los dispositivos Google Nexus y/o Pixel), por lo que Google y otros fabricantes (miembros o no de la *Open Handset Alliance*) pueden seguir procedimientos de actualización diferentes.

194. Finalmente, Google comenzó a publicar con periodicidad mensual la lista oficial de vulnerabilidades de Android para los dispositivos móviles Google Nexus en agosto de 2015, tras la vulnerabilidad conocida como Stagefright. Los boletines de seguridad de Android ("Android Security Bulletins") también se publican en el grupo o foro oficial "Android Security Updates" [Ref.- 69].

²⁹ https://play.google.com/store/devices/details?id=htc_m8 (enlace inactivo a fecha 11/09/2016).

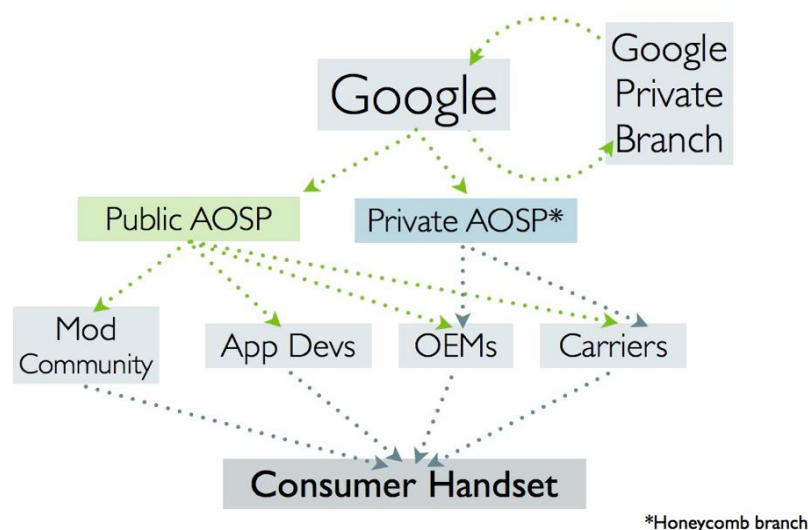
195. Adicionalmente, Google dispone de información complementaria relativa a seguridad de Android y de las vulnerabilidades que pueden afectar a esta plataforma móvil, incluyendo los detalles del proceso recomendado que debe seguirse para su notificación [Ref.- 204].
196. Desde agosto de 2015, tras la vulnerabilidad conocida como Stagefright, Google también inició un programa de publicación y despliegue mensual de actualizaciones de seguridad del sistema operativo Android para sus dispositivos móviles [Ref.- 203], por lo que cada mes se publica una versión nueva de Android, ofreciendo la imagen de fábrica completa (*factory image*) [Ref.- 73] y la versión completa OTA de actualización (*full OTA image*) [Ref.- 199], para los dispositivos móviles bajo soporte.
197. De manera similar, Samsung, otro fabricante de referencia en el ecosistema Android, desde octubre de 2015 ha comenzado a publicar y distribuir boletines de seguridad correspondientes a sus dispositivos móviles Android, también con periodicidad mensual [Ref.- 198].
198. En el caso de los dispositivos móviles Google Nexus, las diferentes versiones del sistema operativo Android son proporcionadas mediante una imagen completa del sistema operativo (o plataforma móvil) Android (ver apartado "6.3. Actualización manual completa de Android"), o como una imagen parcial, normalmente obtenida como actualización OTA (*Over The Air*) (ver apartado "6.4. Actualización manual parcial de Android").
199. Si la imagen es parcial, todos los datos y configuración del usuario son conservados, mientras que, si la imagen es completa, se eliminarán todos los datos y configuración del usuario, fijando los valores de configuración de fábrica durante el proceso. Una imagen completa instala una nueva versión de Android en el dispositivo móvil como si se tratase de un dispositivo recién comprado u obtenido de fábrica.
200. Debe tenerse en cuenta que antes de realizar una actualización del sistema operativo del dispositivo móvil se recomienda realizar una copia de seguridad de los datos del usuario.
201. Pese a que existen foros en Internet donde se distribuyen imágenes de ROM o *firmware* para múltiples sistemas operativos y fabricantes de terminales (como XDA Developers - <http://forum.xda-developers.com>), desde el punto de vista empresarial, en ningún caso se recomienda la instalación del sistema operativo desde una fuente no oficial.
202. En el mundo Android, las actualizaciones individuales adicionales a las actualizaciones de la versión completa del sistema operativo son muy limitadas, en contraposición a las actualizaciones para otros sistemas operativos de sobremesa cliente o servidor. Esta situación afecta tanto a las actualizaciones generales sobre los distintos componentes del sistema operativo, como a las actualizaciones específicas de seguridad.
203. El modelo de soporte actual de los dispositivos móviles comercializados por Google garantiza que los dispositivos Nexus y Pixel recibirán actualizaciones de seguridad durante al menos 3 años desde que el dispositivo empezó a ser comercializado en la Google Store, o al menos durante 18 meses desde su

fecha de fin de comercialización (se toma la fecha más avanzada de las dos), y Google no se compromete a que haya actualizaciones con posterioridad a esa fecha [Ref.- 74].

204. Es importante reseñar que el compromiso de actualizaciones de Google afecta a versiones concretas de Android según el modelo del dispositivo móvil, y no se extiende indefinidamente a la versión original de Android con el que éste se suministró de fábrica (por ejemplo, Android 7.x para el Nexus 5X).
205. Es decir, Google no se compromete a actualizar con los parches de seguridad mensuales todas y cada una de las versiones de Android que han estado disponibles para un dispositivo móvil concreto (Android 6, 7 y 8), sino que actualizará mensualmente únicamente la última versión de Android disponible para los dispositivos Nexus y Pixel aún cubiertos por el soporte de Google. Por ejemplo, la última actualización de seguridad de Android 7.x para el Nexus 5X fue la de agosto de 2017 (N2G48C), ya que los dispositivos Nexus 5X a partir de dicha fecha también disponen de soporte para la versión de Android 8.x.
206. Para el modelo Nexus 5X analizado en la presente guía, la última fecha de compromiso de actualizaciones de seguridad es septiembre de 2018.
207. Desde octubre de 2016, Google añadió la fecha de publicación para cada nueva versión de Android a la nomenclatura empleada en las referencias a los ficheros con las imágenes de fábrica (*factory image*) correspondientes a las actualizaciones de seguridad mensuales. Hasta ese momento, las referencias estaban sólo compuestas por el número de compilación de la versión de Android asociada. El nuevo modelo permite identificar el mes al que corresponde la actualización y, por tanto, el nivel de parche de seguridad de cada fichero.
208. Pese a que, en varias ocasiones, e incluso públicamente, los fabricantes de dispositivos móviles de la *Open Handset Alliance* (OHA) han intentado llegar a un acuerdo y comprometerse a proporcionar actualizaciones durante al menos 18 meses (periodo mínimo de soporte) desde la comercialización inicial de cada modelo de terminal, el tiempo y la historia han confirmado que no han sido capaces de mantener y ratificar ni siquiera estos compromisos públicos.
209. El fabricante del dispositivo móvil o el operador de telefonía es el que publicará actualizaciones para el terminal. Como norma general, y salvo que exista un motivo que desaconseje su instalación, se recomienda aplicar sobre el dispositivo móvil todas las actualizaciones facilitadas por el fabricante, ya que, pese a que las mismas no reflejen explícitamente que se trata de actualizaciones de seguridad, pueden solucionar vulnerabilidades de seguridad públicamente conocidas.
210. Para abordar la comercialización de dispositivos móviles en mercados emergentes, como por ejemplo en la India, asociados con recursos económicos más bajos, Google anunció en el año 2014 la creación de una plataforma denominada Android One, que, en origen, pretendía llegar a

acuerdos con fabricantes de dispositivos móviles de forma que dichos fabricantes desarrollarían terminales de especificaciones y precio reducidos que recibirían soporte directamente por parte de Google. Es decir, en lugar de que el fabricante fuese el encargado de implementar y distribuir las actualizaciones del sistema operativo, Google liberaría las correspondientes imágenes OTA empleando un modelo similar al empleado para los dispositivos Nexus [Ref.- 246].

211. Android One comenzó a estar disponible con Android 4.4.4 (KitKat).
212. Para garantizar las actualizaciones, el programa Android One establece que los dispositivos que se acojan a él deberán incluir únicamente aplicaciones que se puedan gestionar desde Google Play, es decir, sin la capa de aplicaciones propietarias que acompañan comúnmente al hardware de los fabricantes, así como incorporar el servicio Google Play Protect y el asistente de Google (Google Assistant) (consultar apartado "Verificación de seguridad de las apps: Google Play Protect" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]).
213. Con la incorporación al mercado Android One de la compañía Xiaomi y su terminal Mi A1 [Ref.- 247], que, además de un hardware más potente que el que correspondería a los dispositivos móviles objeto del programa inicial, incluye una aplicación de cámara necesaria para aprovechar las capacidades de la cámara dual, Google, en lugar de proporcionar directamente las actualizaciones de sistema operativo a los dispositivos Android One, se encargará de marcar las pautas en relación a las actualizaciones, pero dejará en manos del fabricante el desarrollo y distribución de las mismas de acuerdo a los términos pactados.
214. Google ha desarrollado un modelo para la resolución de problemas de seguridad y la distribución de actualizaciones en Android a través de CTS (*Compatibility Test Suite*) [Ref.- 89], actividad integrada en el *Android Open Source Project* (AOSP), encargada de asegurar la compatibilidad entre dispositivos Android [Ref.- 84]:



215. Sin embargo, toda la complejidad asociada a las actualizaciones de Android y los diferentes factores y entidades involucradas en el proceso, como las múltiples personalizaciones del sistema operativo realizadas por los fabricantes, dan lugar a retrasos significativos desde que las soluciones a problemas de seguridad son añadidas al repositorio del *Android Open Source Project* (AOSP³⁰) hasta que están disponibles para los usuarios, dejando los dispositivos móviles desprotegidos durante todo ese tiempo (varios meses habitualmente, en el mejor de los casos).
216. La siguiente imagen refleja algunas de las vulnerabilidades más significativas en Android durante el periodo entre los años 2010 y 2011, varias de las cuales han sido aprovechadas por código dañino públicamente distribuido, como DroidKungFu y DroidDream³¹:

Known Android Vulnerabilities 2010-2011		
Year	Vulnerability	Malware that Exploited Vuln
2010	Exploid	DroidDream, zHash, DroidKungFu (Legacy)
2010	CVE-2010-1807 WebKit NaN	
2010	RageAgainstTheCage/Zimperlich	DroidDream, BaseBridge, DroidKungFu (Legacy)
2011	Psneuter/KillingInTheNameOf	
2011	Gingerbreak/Honeybomb	

217. Mediante la realización de una búsqueda por el término "Android" en la US CERT NVD (National Vulnerability Database: http://web.nvd.nist.gov/view/vuln/search-results?query=android&search_type=all&cves=on) o en el proyecto CVE (*Common Vulnerabilities and Exposures*) de MITRE (<http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=android>) es posible obtener un listado de las vulnerabilidades de seguridad conocidas públicamente que han afectado a diferentes versiones del sistema operativo Android en los últimos años, así como a aplicaciones de terceros desarrolladas para Android.
218. Adicionalmente, las soluciones (parches o *fixes*) publicadas para corregir vulnerabilidades de seguridad conocidas suelen estar disponibles únicamente para las últimas versiones de Android, no existiendo un mecanismo establecido encargado de proporcionar dichas soluciones para versiones previas de Android, también vulnerables.
219. Por otro lado, numerosos fabricantes y operadores de telefonía móvil no proporcionan actualizaciones para modelos de dispositivos móviles previos, por ejemplo, basados en las versiones previas de Android, como 2.x ó 4.x, por lo que estos serán siempre vulnerables.

³⁰ <https://source.android.com>

³¹ Las referencias que inicialmente publicó Google para informar sobre este ataque, que tuvo enorme repercusión, han sido eliminadas de sus sitios web [Ref.- 88].

220. Este hecho se ve agravado por la velocidad de consumo de las nuevas tecnologías, que afecta a cuándo los dispositivos móviles son considerados obsoletos, donde, en la actualidad, un dispositivo con dos o más años difícilmente estará soportado e incluido en el ciclo de actualizaciones de software.
221. Por ejemplo, para el dispositivo móvil analizado en la presente guía (Google Nexus 5X), que se empezó a comercializar en septiembre de 2015, Google sólo se compromete a garantizar las actualizaciones del sistema operativo hasta septiembre de 2017, y las actualizaciones de parches de seguridad hasta septiembre de 2018.
222. Esta evolución de las versiones de Android asociada a la finalización del periodo de soporte para modelos de dispositivos móviles previos se ha seguido repitiendo a lo largo de los años. Por ejemplo, los dispositivos móviles Nexus 4 y Nexus 10 dejaron de estar soportados cuando se publicó la versión 6.x de Android ("Marshmallow") en octubre de 2015, siendo Android 5.x ("Lollipop") la última versión oficialmente soportada para estos modelos.
223. Asimismo, en agosto de 2016, Google publicó la versión oficial de Android 7.x ("Nougat"), con soporte únicamente para los dispositivos Google Nexus 6, 9, 5X y 6P (junto al Nexus Player, Pixel C y el General Mobile 4G), no soportando dispositivos móviles previos muy extendidos como, por ejemplo, el Nexus 5 y el Nexus 7.
224. La importancia, por un lado, de actualizar la versión de Android en el dispositivo móvil a la última versión disponible, y por otro, de verificar si la actualización corrige vulnerabilidades previas públicamente conocidas, se refleja tanto en vulnerabilidades antiguas relevantes y muy generalizadas como la de TapJacking [Ref.- 116], como en vulnerabilidades críticas más recientes que han afectado a Android, como por ejemplo "Android Master Key" [Ref.- 117], "Android browser SOP bypass" (o Universal XSS, UXSS, en Android < 4.4) [Ref.- 118] (CVE-2014-6041) o sus variantes [Ref.- 100], vulnerabilidades en Android WebView (en Android < 4.4) [Ref.- 119], "Android Fake ID" [Ref.- 101], y otras vulnerabilidades y exploits publicados para Android 4.x y versiones previas hasta el año 2015 [Ref.- 61].
225. Complementariamente, a partir del año 2015, en concreto del mes de agosto, se dispone de la lista oficial de vulnerabilidades que afectan a Android, para sus diferentes versiones a partir de Android 5.x, publicada por Google [Ref.- 69], algunas de ellas de criticidad elevada, como la que originó la publicación mensual de parches de seguridad para Android: Stagefright.
226. En el caso de disponer de un dispositivo móvil Android para el que sí dispone de una actualización a una versión superior de Android, por ejemplo 8.0, para un terminal basado en Android 7.0, se recomienda aplicar la actualización sobre el dispositivo con el objetivo de instalar la última versión disponible y protegerlo de vulnerabilidades conocidas.
227. Con el objetivo de mantener el dispositivo móvil actualizado, se recomienda comprobar periódicamente desde el propio dispositivo la disponibilidad de

actualizaciones desde el menú "Ajustes - Información del teléfono - Actualizaciones del sistema".

228. Desde el punto de vista de seguridad es necesario actualizar igualmente todas las aplicaciones móviles (o apps) instaladas sobre el sistema operativo (ver apartado "20.1. Actualización de apps en Android") y, en particular, la app asociada a los Servicios de Google Play (GPS, *Google Play Services*) debido a las capacidades de actualización que ésta dispone en la propia plataforma móvil Android (ver apartado "6.2. Servicios de Google Play").

6.2. SERVICIOS DE GOOGLE PLAY

229. Durante los últimos años, el concepto de actualización en Android ha cambiado. Muchos componentes de la arquitectura de Android se han independizado del núcleo del sistema operativo de la plataforma móvil, pudiendo ser actualizados independientemente y con mayor frecuencia.
230. Este modelo, que permite que se actualicen de forma individual e independiente algunos componentes de la plataforma Android sin tener que proporcionar una nueva versión del sistema operativo, se lleva a cabo a través de los servicios de Google Play [Ref.- 150].
231. Todos los detalles asociados a dicho servicio, que requiere la existencia de una cuenta de usuario de Google en el dispositivo móvil, pueden consultarse en el apartado "Verificación de seguridad de las apps: Google Play Protect" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407].

6.3. ACTUALIZACIÓN MANUAL COMPLETA DE ANDROID

232. Alternativamente a la actualización automática realizada por Android (ver apartado "6.1. Actualización automática del sistema operativo Android"), o cuando esta opción no está disponible para ciertos fabricantes u operadores de telefonía móvil, es posible realizar la actualización del sistema operativo de Android de forma manual mediante la descarga de los ficheros de actualización oficiales desde la web de Google [Ref.- 73] (o del fabricante).
233. La actualización manual requiere conectar un ordenador mediante USB al dispositivo móvil.
234. En el caso de ciertos fabricantes u operadores de telefonía móvil, en la mayoría de ocasiones estos ficheros no son publicados oficialmente, aunque su ubicación puede estar disponible de manera no oficial desde numerosas fuentes en Internet.
235. Para versiones previas de Android, como por ejemplo 2.x, existía una lista no oficial (incompleta) con los ficheros de actualización proporcionados por Google y por los diferentes fabricantes de dispositivos móviles basados en Android (ver versión previa de la presente guía para Android 2.x [Ref.- 401]).
236. Desde el punto de vista de seguridad, las descargas de actualizaciones siempre deben realizarse desde enlaces que referencien a la web oficial de

- Google (o del fabricante), y es recomendable actualizar siempre a la última versión de Android disponible.
237. En el caso de los dispositivos móviles comercializados por Google, desde hace unos años se proporciona de manera oficial acceso a las imágenes completas de fábrica (*factory images*) para las actualizaciones y nuevas versiones de Android, en concreto, para los dispositivos móviles de la familia Google Nexus y/o Pixels, *Factory Images for Nexus and Pixel Devices* [Ref.- 73] y para Android 7.x [Ref.- 721].
238. De manera complementaria, Google también proporciona oficialmente acceso a las imágenes OTA (Over-the-Air) de fábrica (*full OTA images*) para las actualizaciones y nuevas versiones de Android, en concreto, para los dispositivos móviles de la familia Google Nexus y/o Pixel, *Full OTA Images for Nexus and Pixel Devices* [Ref.- 199]. Es importante reseñar que estas imágenes (ver apartado "6.4. Actualización manual parcial de Android") no son equivalentes a las imágenes parciales que el dispositivo se descarga vía OTA durante una actualización automática (ver apartado "6.1. Actualización automática del sistema operativo Android").
239. En el caso de las actualizaciones automáticas parciales, el dispositivo se descarga un fichero de tamaño más reducido, que sólo incluye los componentes que han cambiado desde la versión de Android instalada actualmente, de la que se parte para la actualización. En el caso de los ficheros o imágenes OTA disponibles a través de Google, el contenido del fichero es casi similar en tamaño al fichero de la imagen completa (descrita en el presente apartado), pues, a diferencia de lo que ocurría en versiones Android 4.x y anteriores (en las que las actualizaciones a una versión requerían partir de otra versión concreta), Google proporciona un único fichero de actualización OTA denominado "Full OTA Image" sea cual sea la versión de Android de partida.
240. La página web oficial de Google [Ref.- 73] con las imágenes completas de fábrica para los dispositivos móviles Google Nexus y/o Pixel se va actualizando periódicamente según se van publicando nuevas versiones de Android (incluyendo los nuevos niveles de parches de seguridad mensuales), pudiendo desaparecer las referencias e imágenes de versiones o subversiones previas de Android.
241. Debe tenerse en cuenta que el proceso de actualización del dispositivo móvil a partir de una imagen de *firmware* completa (*factory image*) eliminará todos los datos del terminal, por lo que se recomienda realizar una copia de seguridad antes de llevar a cabo este proceso.
242. La actualización a partir de una imagen OTA (automática, o manual, *full OTA image*), sin embargo, no borra los datos del usuario, aunque se sigue recomendando realizar una copia de seguridad previa a la actualización.
243. Si la imagen de Android es completa, se dispone de los enlaces a las imágenes oficiales directamente desde Google para los distintos modelos de dispositivos móviles Nexus y/o Pixel, descargables desde "<https://dl.google.com/dl/android/aosp/>". El nombre del fichero ZIP

correspondiente a estas imágenes sigue el siguiente patrón: *nombre clave del tipo de dispositivo móvil + número de compilación + indicativo de que se trata de la imagen completa o de fábrica ("factory") + 8 primeros caracteres hexadecimales del hash SHA256 del fichero ".zip" o ".tgz"*.

Nota: Hasta la versión 4.x de Android, para los ficheros de imagen completos, en lugar de utilizarse el hash SHA256 se usaba el hash SHA1.

Nota: Para los ficheros de actualización automática (OTA), en lugar del SHA256 Google sigue utilizando internamente el SHA1 en su nomenclatura.

244. Por ejemplo, el fichero de la imagen de fábrica completa correspondiente a la versión Android 7.1.2 (N2G47O, de mayo de 2017) empleada para la elaboración de la presente guía en un terminal Nexus 5X ("bullhead"), cuyo valor SHA256 es 3ff32f55daae9d973021c3e67480d3210ace7aabc48eede762d2cea29180e3b2:

<https://dl.google.com/dl/android/aosp/bullhead-n2g47o-factory-3ff32f55.zip>

245. Por otro lado, el fichero de la imagen OTA parcial correspondiente a la versión Android 7.1.2 (N2G47O, de mayo de 2017) empleada para la elaboración de la presente guía en un terminal Nexus 5X ("bullhead"), cuyo valor SHA256 es b6cdc215d284184b8a8a01a7c480f8698eb5c96f21735de73235a3456e9f7b50, sería:

<https://dl.google.com/dl/android/aosp/bullhead-ota-n2g47o-b6cdc215.zip>

246. Antes de iniciar el proceso de actualización del dispositivo móvil, es necesario verificar que el nivel de batería del dispositivo es suficiente (es decir, superior al 50%).

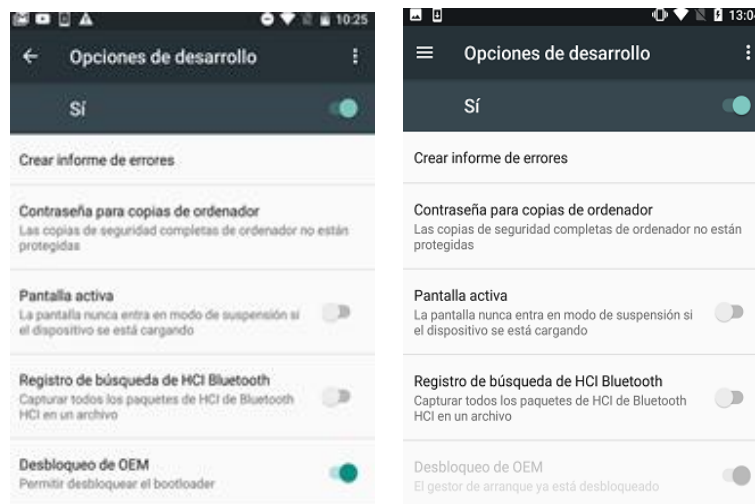
247. También es necesario hacer uso de la herramienta "fastboot", disponible como parte del Android SDK, y también de forma independiente³². Las "platform tools" de Android incluyen además la herramienta "adb", que permite acceso vía USB al dispositivo móvil como desarrollador mediante el protocolo ADB³³ (*Android Debug Bridge*); ver apartado "13.3. Opciones de desarrollo y depuración USB" [Ref.- 110].

248. Para llevar a cabo el proceso de actualización completa desde una imagen de fábrica (*factory image*), desde Windows, Linux o macOS, es necesario desbloquear el gestor de arranque (o *bootloader*) del dispositivo móvil Android, si aún no ha sido desbloqueado previamente.

³² <https://developer.android.com/studio/releases/platform-tools.html>

³³ Para utilizar el protocolo ADB con el objetivo de interactuar con el dispositivo móvil y proceder a actualizar la versión de Android, es necesario descargar la última versión de las "platform tools" del Android SDK (la versión mínima necesaria es la 25.0.5, ya que, en versiones anteriores, el demonio de "adb" abortaba su ejecución para ficheros de gran tamaño; consultar "https://issuetracker.google.com/issues/37139736" para más detalles).

249. A partir de Android 6.x, para desbloquear el gestor de arranque, es necesario adicionalmente habilitar la opción "Ajustes - Opciones de desarrollo - Desbloqueo de OEM", que no existía en versiones iguales o previas a Android 5.x. Esta operación requiere que el usuario introduzca el código de acceso (PIN, patrón o contraseña) actualmente empleado como método de protección de acceso al dispositivo móvil. Para poder activar esta opción, es preciso previamente habilitar las opciones de desarrollo (ver apartado "13.3. Opciones de desarrollo y depuración USB").

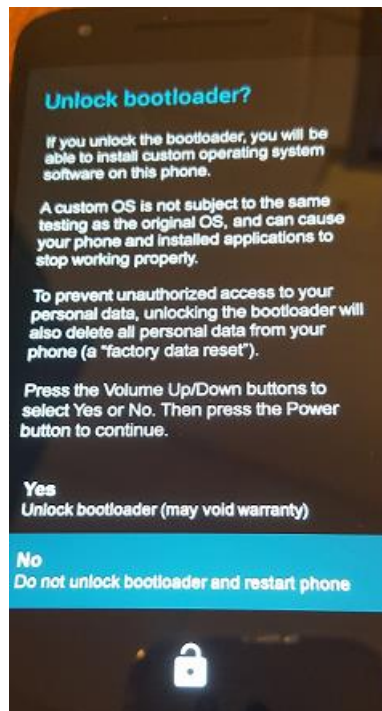


250. Si el gestor de arranque se hubiese desbloqueado previamente, desde el menú "Ajustes - Opciones de desarrollo", se mostrará el valor "Desbloqueo OEM - el gestor de arranque ya está desbloqueado". Una vez seleccionada la opción "Ajustes - Opciones de desarrollo - Desbloqueo de OEM", es posible proceder a desbloquear el *bootloader* mediante dos métodos diferentes equivalentes:

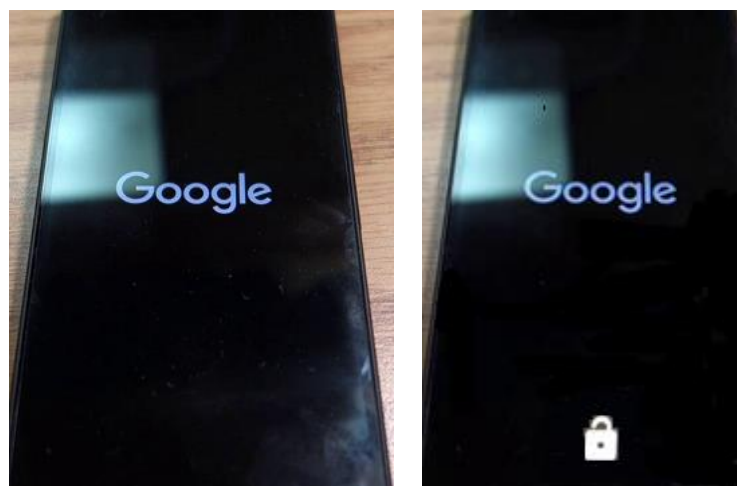
- Apagando el dispositivo móvil y rearrancando el mismo con el botón de bajar el volumen y el botón de encendido pulsados simultáneamente hasta que aparezca la pantalla del *bootloader*. Si el valor de "Device state" del dispositivo aparece como "locked", indica que el gestor de arranque está bloqueado. Para desbloquearlo, desde el ordenador conectado por USB al dispositivo móvil se deberá ejecutar el comando "*fastboot flashing unlock*" (hasta Android 5.x incluido, el comando era "*fastboot oem unlock*", que sigue vigente en Android 6.x y 7.x pero cuyo uso se desaconseja frente al nuevo comando). En el dispositivo móvil, se mostrará una pantalla informando de que esta operación eliminará todos los datos del dispositivo, y solicitará confirmación por parte del usuario para continuar.
- Utilizando el protocolo ADB, y la herramienta asociada "adb", desde el ordenador conectado por USB al dispositivo móvil. Previamente, habrá que habilitar las opciones de desarrollo mediante "Ajustes - Información del teléfono", pulsando 7 veces sobre el "Número de compilación". Esta operación hará que aparezca un nuevo menú "Ajustes - Opciones de desarrollo" y, dentro de éste, la opción "Depuración por USB". Una vez activada esta opción, quedará habilitado el acceso USB al dispositivo móvil como desarrollador mediante el protocolo ADB (*Android Debug Bridge*)

[Ref.- 110] (ver apartado "13.3. Opciones de desarrollo y depuración USB"). Estas capacidades permiten que se pueda ejecutar el comando "*adb reboot bootloader*" para arrancar desde el *bootloader*, acceder al gestor de arranque y, una vez en él, poder ejecutar el comando "*fastboot flashing unlock*" para desbloquear el gestor de arranque en caso de estar bloqueado.

251. Tras recibir la confirmación del usuario (utilizando el botón de subir volumen y presionando el botón de encendido) a través de la pantalla del dispositivo móvil, se eliminarán todos los datos existentes en el dispositivo móvil y se desbloqueará el *bootloader*.



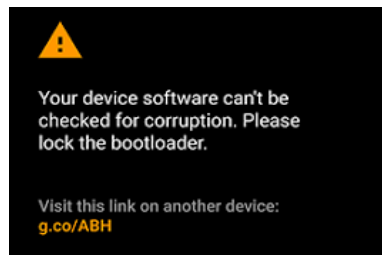
252. Cuando el *bootloader* está bloqueado, en el arranque del terminal no se muestra ningún tipo de candado, a diferencia de lo que sucedía en versiones anteriores a Android 6.x. Si se desbloquea el *bootloader*, en la pantalla de arranque se mostrará un candado abierto en la parte inferior central del dispositivo móvil:



253. Cuando el *bootloader* está desbloqueado, la pantalla del gestor de arranque o modo *fastboot* muestra el valor "Device State - unlocked" (en versiones anteriores de Android, se mostraba "Lock State - unlocked").



254. El desbloquear el *bootloader*, cuando se reinicia el dispositivo móvil se mostrará el siguiente mensaje:



255. Esto es debido al mecanismo "Verified boot" (consultar apartado "7.9. Proceso de arranque seguro: Verified Boot"), responsable de examinar el código del sistema operativo al encender un dispositivo Android para verificar su integridad y confirmar que no ha sido modificado y que se puede utilizar de forma segura.
256. Una vez desbloqueado el *bootloader*, el dispositivo móvil permanecerá en este estado aunque se borre la partición de sistema o de datos, por ejemplo, tras llevar a cabo una restauración a fábrica.
257. Seguidamente, se ha de conectar el dispositivo móvil a través de USB y acceder al gestor de arranque (o *bootloader*), también conocido como modo *fastboot* (para dispositivos móviles Nexus y/o Pixel). De nuevo, en caso de haber empleado el método de desbloqueo del gestor de arranque descrito previamente que hacía uso del protocolo ADB, se puede emplear el comando "*adb reboot bootloader*" (a través de USB), o el propio dispositivo móvil (apagando el dispositivo y encendiéndolo mientras se mantienen pulsados simultáneamente el botón de encendido y el botón de bajar el volumen, para el caso concreto del dispositivo móvil empleado para la elaboración de la presente guía).
258. Una vez descargada la imagen completa del *firmware* de la versión de Android a instalar, debe descomprimirse el fichero ZIP, y acceder al directorio asociado recién creado:

```
C:\software\versiones Android\7_0_0_bullhead\bullhead-n5d911 > dir
Directorio: C:\software\versiones Android\7_0_0_bullhead\bullhead-n5d911
Mode      LastWrite Time         Length Name
----      -
-a----- 06/04/2018 10:15      4714228 bootloader-bullhead-bhz11g.img
-a----- 06/04/2018 10:14         968 flash-all.bat
-a----- 06/04/2018 10:14         839 flash-all.sh
-a----- 06/04/2018 10:14         794 flash-base.sh
-a----- 06/04/2018 10:15    1213513318 image-bullhead-n5d911.zip
-a----- 06/04/2018 10:14     58120704 radio-bullhead-m8994f-2.6.33.2.14.img
```

259. Mediante la ejecución del script "flash-all" (".bat" o ".sh", según se haga uso de un ordenador basado en Windows o en Linux) se llevará a cabo la instalación de la versión de Android correspondiente, incluyendo la instalación de la imagen del gestor de arranque (*bootloader*), del *firmware* del procesador de banda base (*baseband*) o radio, y del sistema operativo Android o partición de sistema ("/system").
260. Si el *bootloader* no está desbloqueado, aunque se ejecute el script "*flash-all*" y aparentemente se estén realizando las acciones contempladas en el mismo, el script terminará fallando con el error "*Failed (remote: device is locked. Cannot flash images)*".
261. Una vez el dispositivo móvil ha sido actualizado (tras ejecutar el script y pasados unos minutos), se reiniciará y arrancará la nueva versión de Android aplicada. El dispositivo móvil arrancará en su estado original de fábrica, por lo que será necesario completar el proceso de instalación y configuración inicial (ver apartado "23. Apéndice A: Proceso de instalación y configuración inicial").

Nota: El proceso de actualización desde una imagen completa de Android mediante el script "*flash-all*" también elimina por defecto todos los datos de usuario del dispositivo móvil. Esta es la opción de actualización recomendada con el objetivo de evitar problemas de compatibilidad entre diferentes versiones de Android debido a la existencia de contenidos y datos previamente existentes. Sin embargo, existe la posibilidad de modificar el script "*flash-all*" y en concreto la directiva "*fastboot -w update ...*", eliminando la opción "-w" (*wipe*), para que los datos de usuario no sean borrados durante el proceso de actualización.

262. Desde el punto de vista de seguridad se recomienda bloquear de nuevo el gestor de arranque (*bootloader*) nada más completar el proceso de actualización. Para ello, basta con apagar el dispositivo móvil tras el primer reinicio después de la actualización, arrancar de nuevo en modo *fastboot* mediante ADB a través de USB o desde el propio dispositivo móvil con la combinación de botones descrita previamente, y conectar el dispositivo móvil a un ordenador mediante el puerto USB. Desde el mismo, es posible ejecutar el comando "*fastboot flashing lock*" para volver a bloquear

inmediatamente el gestor de arranque. El estado es indicado en la pantalla del propio gestor de arranque mediante el mensaje ("Device State - locked"):



263. A partir de Android 6.x, si se hubiese llevado a cabo alguna personalización antes de volver a bloquear el *bootloader*, sería necesario realizar una copia de seguridad del dispositivo móvil (ver apartado "21. Copias de seguridad"), ya que el proceso conlleva una restauración a fábrica con la consecuente eliminación de todos los datos de usuario. Este no era el comportamiento de las versiones de Android 5.x o anteriores, en las que el proceso de bloqueo del gestor de arranque no eliminaba ningún dato del dispositivo.
264. Por tanto, una vez bloqueado de nuevo el gestor de arranque (o *bootloader*), será necesario completar el proceso de instalación y configuración inicial.
265. De esta forma, una vez el *bootloader* está bloqueado, no será posible actualizar el dispositivo móvil mediante una actualización completa, ni interactuar con las diferentes particiones existentes en el dispositivo, sin tener que eliminar previamente todos los datos del usuario.

6.4. ACTUALIZACIÓN MANUAL PARCIAL DE ANDROID

266. En el caso de los dispositivos móviles Google Nexus y/o Pixel, las diferentes versiones del sistema operativo Android son proporcionadas mediante una imagen completa del sistema operativo (o plataforma móvil) Android (ver apartado "6.3. Actualización manual completa de Android"), o como una actualización parcial, normalmente obtenida como actualización OTA (*Over-the-Air*) automáticamente por parte de Android (ver apartado "6.1. Actualización automática del sistema operativo Android").
267. Alternativamente a la actualización automática realizada directamente por Android vía OTA (ver apartado "6.1. Actualización automática del sistema operativo Android"), o al proceso de instalación o actualización manual desde una imagen completa de *firmware*, *factory image* (ver apartado "6.3. Actualización manual completa de Android"), a partir de Android 6.x es

posible realizar la actualización del sistema operativo de Android de forma manual mediante la descarga de los ficheros de actualización OTA completos, o imágenes OTA completas (*full OTA images*), oficiales desde la web de Google (o del fabricante).

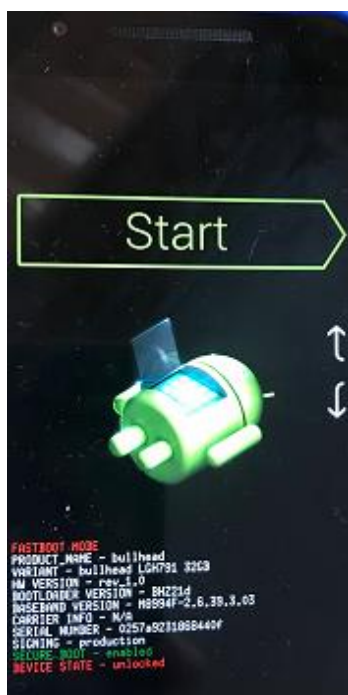
268. Estos ficheros no son los mismos que se descarga el dispositivo móvil mediante el mecanismo OTA durante una actualización automática, pues las actualizaciones automáticas detectan la versión de partida del dispositivo móvil y le envían únicamente un fichero que contiene los cambios correspondientes a la versión de Android a la que se actualizará.
269. La principal diferencia entre una actualización manual completa y una actualización manual parcial a través de una imagen OTA es que ésta última mantiene los datos de usuario.
270. En versiones de Android 5.x y anteriores, las actualizaciones parciales solo eran distribuidas a través de OTA de forma automática con conexión a Internet, y no se disponía de los enlaces a las imágenes OTA completas oficiales directamente desde Google para los distintos modelos de dispositivo móvil. Actualmente, se dispone de las imágenes OTA completas para versiones iguales o superiores a Android 6.0.1. Las imágenes completas (no OTA) sí están disponibles para versiones de Android desde la versión 2.x.
271. Adicionalmente, por un lado, existen repositorios no oficiales con referencias, en concreto para los dispositivos móviles de la familia Google Nexus (de interés y aplicación en la presente guía) o Google Experience Devices, que permiten identificar los diferentes ficheros correspondientes a las actualizaciones parciales disponibles, al igual que obtener las referencias a los enlaces OTA oficiales de Google utilizados de manera automática por Android, desde foros como XDA Developers³⁴.
272. Por otro lado, mediante el análisis de los registros o logs de logcat, o del tráfico de red del proceso de actualización de Android, es posible obtener los enlaces a los ficheros de descarga de las actualizaciones OTA, que están disponibles desde el servidor "ota.googlezip.net/packages/ota-api/*". Es importante reseñar que estos ficheros no están disponibles para su descarga oficial desde los servidores de Google, por lo que no se recomienda su aplicación.
273. Por otro lado, mediante el análisis de los registros o logs de logcat, o del tráfico de red del proceso de actualización de Android, es posible obtener los enlaces a los ficheros de descarga de las actualizaciones OTA, que están disponibles desde el servidor "https://ota.googlezip.net/packages/ota-api/*". Es importante reseñar que estos ficheros no están disponibles para su descarga pública oficial desde los servidores de Google, por lo que no se recomienda su aplicación directa. Por ejemplo, la actualización a la versión Android 7.1.2 con el parche de seguridad de mayo de 2017 (N2G47O), desde la versión Android 7.1.2 con el parche de seguridad de marzo de 2017 (N2G47F), empleada para la elaboración de la presente guía en un dispositivo móvil Nexus 5X ("bullhead"), puede obtenerse desde:

³⁴ <https://forum.xda-developers.com>


```
https://ota.googlezip.net/packages/ota-api/google_bullhead_
bullhead/a9f4fe0887c33f50db027480550c7b2ec49e231a.zip
```

274. Estos ficheros resultan de menor tamaño que las imágenes OTA completas (*full OTA images*) que se pueden descargar manualmente desde el sitio web oficial de Google, ya que contienen sólo los cambios a aplicar entre la versión actual y la versión de Android destino tras aplicarse la actualización.
275. El nombre del directorio incluye el término "google_", seguido del nombre en clave del tipo de dispositivo móvil, en este caso "bullhead" (por duplicado) asociado al Nexus 5X. El nombre del fichero en este caso no corresponde al valor (o parte del valor) del hash SHA256 del fichero ZIP de la actualización, como ocurre con las versiones de las *full OTA images* o de las actualizaciones completas (*factory images*), sino al valor del hash SHA1 de dicho fichero ZIP. Este formato basado en el valor del hash SHA1 ya se utilizaba en la nomenclatura de las actualizaciones parciales en versiones de Android 4.x y anteriores.
276. El proceso de actualización manual parcial (desde Windows, Linux o macOS) requiere disponer del fichero oficial de actualización OTA completo descargado desde la web de Google, y es también necesario disponer del SDK de Android y habilitar el acceso por USB al dispositivo móvil como desarrollador, en concreto, mediante ADB, al igual que para realizar una actualización manual completa. Se recomienda verificar el hash del fichero de la imagen para asegurarse de que es correcto, comparando los 8 primeros caracteres hexadecimales del valor SHA256 (en el caso de imágenes *full OTA image* descargadas desde la página oficial de Google) o SHA1 (para imágenes OTA parciales que se han descargado mediante el análisis del tráfico o logs) con los disponibles en el nombre del fichero.
277. En el caso de Android 5.x, este método permite únicamente la actualización de la versión de Android disponible en el dispositivo móvil mediante una imagen OTA parcial no oficial.
278. A partir de Android 6.x, este método también permite la actualización de la versión de Android disponible en el dispositivo móvil mediante una imagen OTA completa oficial (*full OTA image*).
279. Una vez el dispositivo móvil se ha conectado al ordenador mediante el puerto USB, es necesario reiniciarlo desde la partición de recuperación (o *recovery*). Para ello se puede utilizar el comando "adb" desde el ordenador vía USB, el gestor de arranque (accesible mediante la combinación adecuada de botones en el dispositivo móvil para acceder al *bootloader*, tal como se ha descrito anteriormente) o acceder directamente mediante una nueva combinación de botones.
280. Para acceder a la imagen de recuperación (o *recovery*) mediante el comando "adb" es necesario disponer de acceso como desarrollador mediante USB, y emplear el comando "*adb reboot recovery*", que mostrará la pantalla de la imagen de recuperación con un androide y un icono de peligro (descrita posteriormente).

281. Para acceder a la imagen de recuperación a través del gestor de arranque o *bootloader*, ya que éste dispone de una opción de menú de recuperación específica (en el dispositivo móvil empleado para la elaboración de la presente guía), puede emplearse el comando "*adb reboot bootloader*" desde el ordenador mediante USB. A continuación, debe mantenerse pulsado el botón de encendido y posteriormente pulsar el botón de subir el volumen simultáneamente hasta que aparezca el mensaje "*Recovery mode*" en la pantalla. o se puede acceder al *bootloader* desde el propio dispositivo móvil³⁵, mediante la combinación de botones correspondiente (tal y como se describió en el apartado "6.3. Actualización manual completa de Android"). Una vez en el gestor de arranque, es posible arrancar desde la imagen de recuperación desplazándose hasta la opción "*Recovery mode*" mediante el botón de bajar volumen y seleccionándola mediante el botón de encendido.
282. Para acceder directamente a la imagen de recuperación (o *recovery*) desde el propio dispositivo móvil, se dispone de una combinación de botones específica. Se debe apagar el dispositivo móvil y encenderlo (en el caso del dispositivo móvil empleado para la elaboración de la presente guía) manteniendo pulsados los botones de bajar y subir el volumen simultáneamente. El dispositivo móvil se reiniciará y mostrará una pantalla de recuperación que contiene un androide y el menú "Start". Se pulsará entonces el botón de bajar volumen hasta que aparezca el menú "*Recovery mode*" en la pantalla.



283. En cualquiera de los casos, el dispositivo se reiniciará y aparecerá en pantalla la imagen de un androide con el símbolo de un triángulo que contiene una admiración de color rojo (⚠) y el texto "*No command*".

³⁵ <https://source.android.com/source/building-devices.html#booting-into-fastboot-mode>



284. Para acceder a las opciones del menú de recuperación, será necesario presionar mantener pulsado el botón de encendido y posteriormente pulsar el botón de subir el volumen simultáneamente hasta que aparezca el menú de recuperación.



285. El modo de recuperación es mostrado en la pantalla con el texto "Android Recovery" en color amarillo en la parte superior del dispositivo móvil y, debajo, la línea "Google/bullhead/bullhead", seguida de otra línea que incluye la versión actual de Android que disponible en el dispositivo y el número de compilación de dicha versión. Con el botón de bajar volumen es posible desplazarse a la opción "Apply update from ADB", debiendo usar el botón de encendido para seleccionar la opción deseada.
286. Cuando el dispositivo se encuentra ejecutando la imagen de recuperación, y se habilitan las capacidades de comunicación mediante ADB, el estado del dispositivo es diferente al empleado en el acceso ADB estándar, y está identificado por el término "sideload". Así, al ejecutar el comando "adb devices" en el ordenador, se verá el número de serie asociado al dispositivo móvil seguido del texto "sideload".
287. Para llevar a cabo una actualización manual parcial en el dispositivo móvil, tal como ocurre con las actualizaciones automáticas OTA, o una actualización manual mediante una imagen OTA completa, no es necesario desbloquear el gestor de arranque (o *bootloader*).
288. A continuación, es necesario ejecutar desde el ordenador el comando que lleva a cabo la carga del fichero de actualización parcial a través del puerto USB: "adb sideload <nombre_de_fichero>.zip", donde "<nombre_de_fichero>.zip" corresponde al nombre del fichero ZIP con la actualización parcial a aplicar. Este comando debe ejecutarse directamente desde el directorio donde se descargó dicho fichero.
289. Desde Android 6.x, este procedimiento sirve tanto para aplicar una imagen OTA parcial no oficial, como una imagen OTA completa oficial (*full OTA image*), mostrada a continuación:

```
C:\Android> adb devices
List of devices attached
* daemon not running. starting it now at tcp:5037 *
* daemon started successfully *
0257a92318987654      sideload

C:\Android> adb sideload 7.1.2_bullhead-ota-n2g47o-b6cdc215.zip
opening '7.1.2_bullhead-ota-n2g47o-b6cdc215.zip'...
connecting...
serving: '7.1.2_bullhead-ota-n2g47o-b6cdc215.zip' (~18%)
Total xfer: 2.00x
```

Nota: En algunos casos (dependiendo de las modificaciones realizadas por la actualización y de la versión exacta previa y la versión final de Android) puede ser recomendable adicionalmente, una vez se ha completado la misma, y desde la imagen de recuperación, limpiar la caché del dispositivo móvil a través de la opción del modo de recuperación del gestor de arranque "*wipe cache partition*".

290. Una vez se ha completado la instalación de la actualización, estado identificado por el mensaje "*Install from ADB complete.*", se debe seleccionar la opción "*reboot system now*" para reiniciar el dispositivo móvil con la nueva actualización aplicada, proceso que puede llevar algún tiempo.

291. Cuando el dispositivo móvil ha sido actualizado, tras reiniciarlo, ejecutará la nueva versión de Android, conservando los datos de usuario existentes previamente, tal como ocurriría con una actualización automática recibida mediante OTA (vía Wi-Fi o 2/3/4G).

292. En Android 7.x, se ha reducido el tiempo necesario de instalación de las actualizaciones OTA. No es necesario esperar a que se lleve a cabo la optimización de las apps (la parte más lenta del proceso de actualización), sino que el nuevo compilador JIT (*Just in Time*) ha sido optimizado para hacer que el proceso de instalación y actualización sea mucho más rápido [Ref.- 706].

293. Por otro lado, las actualizaciones de sistema operativo en Android 7.x se llevan a cabo mediante un proceso de fondo (o *background*), mientras el dispositivo móvil continúa su ejecución, como se detalla a continuación.

6.5. A/B (SEAMLESS) SYSTEM UPDATES

294. Android 7.x (Nougat) introduce como una de sus principales novedades un esquema de actualizaciones de sistema operativo que recibe el nombre de "A/B (Seamless) System Updates" (o actualizaciones sin interrupciones, también conocido como "seamless updates"), y que es de aplicación únicamente para los nuevos modelos de dispositivo móvil, es decir, aquellos que se empezaron a comercializar con posterioridad a la publicación de Android 7.0 (en agosto de 2016) y que traían esta versión preinstalada de fábrica [Ref.- 726].

295. Nótese que los dispositivos Nexus 5X (como el empleado para la elaboración de la presente guía), no entran pues en el grupo de dispositivos que

soportan actualizaciones sin interrupciones, ya que su comercialización se inició en septiembre de 2015. El hecho de que los modelos más recientes de Nexus 5X viniesen con Android 7.x de fábrica hizo pensar que quizá soportarían este tipo de actualización, pero, aunque Google no proporcionó justificaciones al respecto, la creencia extraoficial es que el reparticionamiento del hardware que requiere el nuevo modelo de actualizaciones no era viable.

296. Por tanto, el primer dispositivo fabricado por Google que soportó las actualizaciones sin interrupciones fue el Pixel (y Pixel XL), comercializado en octubre de 2016.

297. A fecha de elaboración de la presente guía, existe una lista no oficial de dispositivos que soportan el modelo "A/B Seamless System Updates" [Ref.- 733].

298. Este nuevo esquema de actualizaciones de sistema se asienta sobre un conjunto de particiones lógicas dual, cada uno de los cuales se denomina *slot* en terminología Android, de forma que un slot será el activo (online) y el otro estará en segundo plano (offline). Cada slot está integrado por un conjunto de particiones lógicas necesarias para el correcto funcionamiento del sistema. Por tanto, para el proceso de actualización sin interrupciones, se crea un segundo slot en el almacenamiento del dispositivo móvil, que complementa al existente en las versiones previas de Android. Cada slot consta de la partición de arranque (boot), la de sistema (system) y una específica para el fabricante (vendor) [Ref.- 725].

299. Estos slots se referencian como "a" y "b", de ahí el nombre del esquema de actualizaciones "A/B Seamless System Updates".

300. Es posible averiguar si el dispositivo móvil soporta el esquema mediante los siguientes comandos "adb":

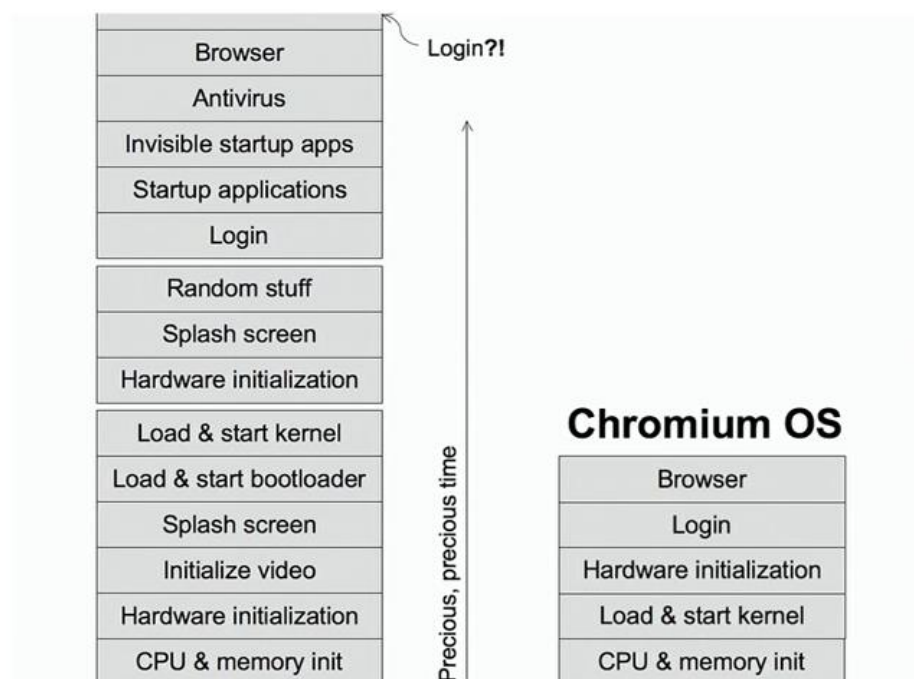
```
C:\Android> adb shell
bullhead:/ $ getprop ro.build.ab_update
Este comando devolverá valor "true" si el dispositivo soporta el esquema. Si no devuelve nada, indica que no lo soporta.

bullhead:/ $ getprop ro.boot.slot_suffix
Este comando devolverá valor "_a" o "_b" si el dispositivo soporta el esquema. En caso contrario, no devuelve nada.
```

301. Los cambios de cara al sistema de particiones son:

- Desaparecen las particiones de recuperación (recovery) y caché.
- El directorio raíz ("/") deja de formar parte de la partición de arranque y se integra en la partición de sistema. La partición de sistema contiene el "initramfs" (init y ramdisk), que se monta en "/" (rootfs) en lugar de en "/system".
- La partición de recovery se integra en la imagen o partición de arranque (boot), dentro de un ramdisk.

- La partición de datos de usuario ("/data", que contiene todos los datos de usuario, las apps descargadas y los datos que dichas apps hayan generado) no se duplica, es decir, sólo existe una versión de dicha partición cifrada (permanece igual que en el esquema de versiones previas de Android).
302. Este esquema de slots (conjunto de particiones) permite que se puedan realizar actualizaciones de sistema operativo sobre la partición que no se encuentra activa (offline) mientras que el usuario interactúa de forma normal con el dispositivo móvil a través del slot activo (online). Esto supone una gran mejora de cara a las actualizaciones, pues en versiones anteriores de Android el dispositivo no estaba operativo mientras se completaba el proceso de actualización de sistema, lo cual podía llevar hasta 25-30 minutos en el caso de algunos terminales³⁶. Este aspecto es especialmente importante teniendo en cuenta que en la actualidad Google ofrece actualizaciones de seguridad mensualmente.
303. El concepto de "seamless updates" procede de Chromium OS, un sistema operativo de código abierto que prescinde de muchos de los componentes de la arquitectura de un sistema operativo tradicional, pensado para usuarios que utilizan fundamentalmente recursos web. La imagen inferior ilustra estas diferencias de componentes³⁷:



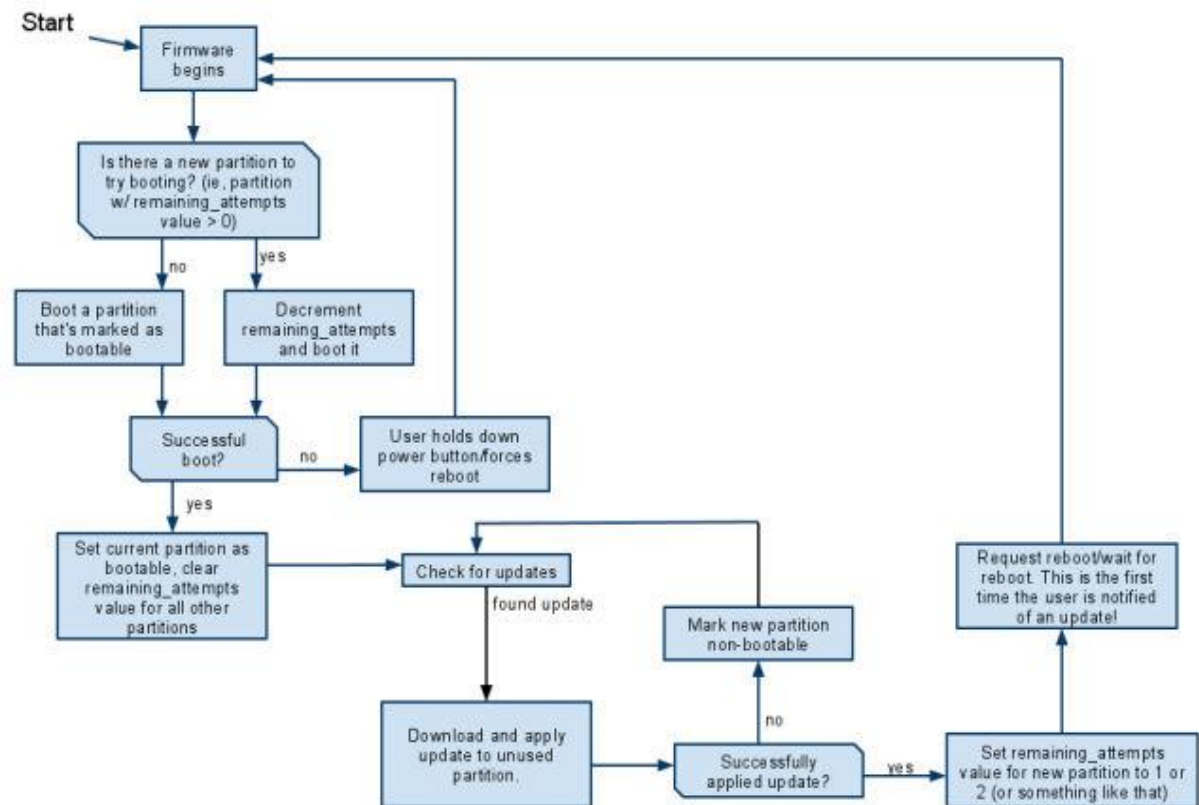
304. Cuando el dispositivo se reinicia después de una actualización de sistema (que ha tenido lugar en segundo plano en la partición no activa), Android intercambiará el estado de los slots, de modo que el slot sobre el que se llevó a cabo la actualización (que se encontraba offline) pasará a ser el activo (online) y viceversa. Por tanto, la actualización del sistema operativo sólo requerirá un reinicio. Cada slot tiene asociado un atributo que define si el

³⁶ https://cdn.arstechnica.net/wp-content/uploads/2016/05/Screenshot_20160513-200938.png

³⁷ <https://www.youtube.com/watch?v=mTFfI7AjNfI>

slot contiene una imagen de sistema correcta de la que se podrá arrancar. Así, el slot actualmente en ejecución tiene el atributo que lo identifica como "arrancable" (bootable), mientras que el segundo slot (no activo) puede tener una versión más antigua (pero correcta) del sistema, una más reciente o datos inválidos.

305. Además de la ventaja que supone que la actualización se realice sin intervención del usuario y sin requerir tiempo de inactividad, las "seamless updates" evitan que, en caso de que la actualización sea fallida por cualquier causa, el dispositivo móvil quede inutilizado o inoperativo, y sea preciso restaurarlo a fábrica.
306. Tras la actualización, el gestor de arranque asignará un valor "*remaining_attempts*" (número de intentos) al slot actualizado, que se almacenará en la tabla de particiones junto al flag "bootable" (que es el que determina si la partición es de arranque). Cuando se reinicie el sistema, el bootloader examinará las particiones existentes en el sistema e intentará arrancar de aquella que tenga un "*remaining_attempts* > 0". El valor se decrementará y el sistema arrancará desde esa partición; si se produce un fallo, se repetirá el proceso hasta que "*remaining_attempts*" llegue a 0, en cuyo caso se abandonará el intento de arranque sobre esa partición y se arrancará de otra marcada como partición de arranque (que, en el caso de Android 7.x, será la que existía en el sistema antes de aplicarse la actualización del sistema operativo). Actualmente el valor de *remaining_attempts* que se asigna a la partición actualizada es de 6.
307. El siguiente diagrama ilustra el proceso de arranque tras una actualización, correspondiente a la arquitectura de Chromium OS, ya que Android emplea un mecanismo similar [Ref.- 715]:



308. Las actualizaciones se aplican por tanto sobre la otra partición de sistema, y en el siguiente arranque del dispositivo, se comienza a hacer uso de esa nueva partición con el sistema operativo actualizado de manera transparente.

309. Otra de las ventajas que presenta este esquema es que protege al dispositivo móvil frente a ciertas amenazas y *exploits*, ya que la partición activa de sistema se puede montar en modo de sólo lectura (*read-only*) [Ref.- 706], puesto que ni tan siquiera el proceso de actualización requerirá escribir sobre ella.

310. Adicionalmente, cada bloque de almacenamiento físico de la partición de sistema se firmará para garantizar su integridad, tal como se describe en el apartado "7.9. Proceso de arranque seguro: Verified Boot".

311. Es importante reseñar que el modelo de actualización sin interrupciones es válido para los dispositivos basados en actualizaciones mediante mecanismos OTA (Over-The-Air). En la infraestructura de Google de actualizaciones OTA, Google proporciona una parte cliente (o app) y una parte de sistema dentro del propio Android para realizar las actualizaciones OTA. Los cambios necesarios para "seamless updates" en la parte de sistema se proporcionan a través del AOSP (en el propio código de Android 7.x), y el código cliente se suministra a través de la app Google Play Services. Aquellos fabricantes (OEMs) que no emplean la infraestructura de Google para las actualizaciones OTA, podrán hacer uso directamente del código de sistema de Android del AOSP, pero tendrán que proporcionar su propio software cliente (o app) si desean que sus dispositivos se beneficien de este modelo.

312.El nuevo modelo de slots duales de Android 7.x conlleva también cambios en la utilidad fastboot, algunos relativos a opciones existentes previamente y otros relativos a opciones nuevas:

- Opción "*update <fichero.zip>*": lanza un proceso de actualización partiendo del fichero de actualización indicado como parámetro y marca el slot actualizado como activo.
- Opción "*flashall*": si el dispositivo soporta slots, el slot sobre el que se haga el proceso de volcado (*flash*) de la nueva imagen se marca como activo. Las imágenes secundarias pueden ser volcadas sobre un slot inactivo. Para mantener el slot inactivo intacto y arrancable aunque el comando "*flashall*" o el comando "*update*" conllevasen una modificación del slot no activo, se deberá añadir el parámetro "*--skip-secondary*".
- Opción "*set_active <slot>*" o "*--set-active=<slot>*": fija el slot activo si el dispositivo soporta slots. Básicamente cambia el flag del bootloader, aunque esto no implica que el nuevo slot se convierta en activo en ese momento (como en el caso de estarse ejecutando el bootloader sobre el slot inactivo). Es importante reseñar que, si se marca como activo un slot y el dispositivo no puede arrancar de él, intentará la operación hasta que se agote el valor "*remaining_attempts*" y, si no tiene éxito, cambiará al otro slot.
- Opción "*--slot <slot>*": esta opción se introduce en Android 7.x para fijar el slot objetivo. Admite los valores '*all*' (con ello las operaciones se aplicarán sobre ambos slots) y '*other*' (para que las operaciones se apliquen al slot no activo).

313.Si el dispositivo no soporta slots, cualquier opción de fastboot de las mencionadas anteriormente será ignorada o mostrará un mensaje de error:

```
C:\> fastboot set_active _b
error: Device does not support slots.
```

6.5.1. PROCESO DE ACTUALIZACIÓN SEAMLESS UPDATE

314.El proceso de actualización se inicia cuando el paquete OTA (al que el código Android referencia como *payload*) se libera y las políticas del dispositivo permiten su procesamiento (por ejemplo, el estado de la batería o la actividad que en ese momento esté llevando a cabo el usuario, pueden diferir la descarga e instalación del denominado *payload*). Un *payload* contiene:

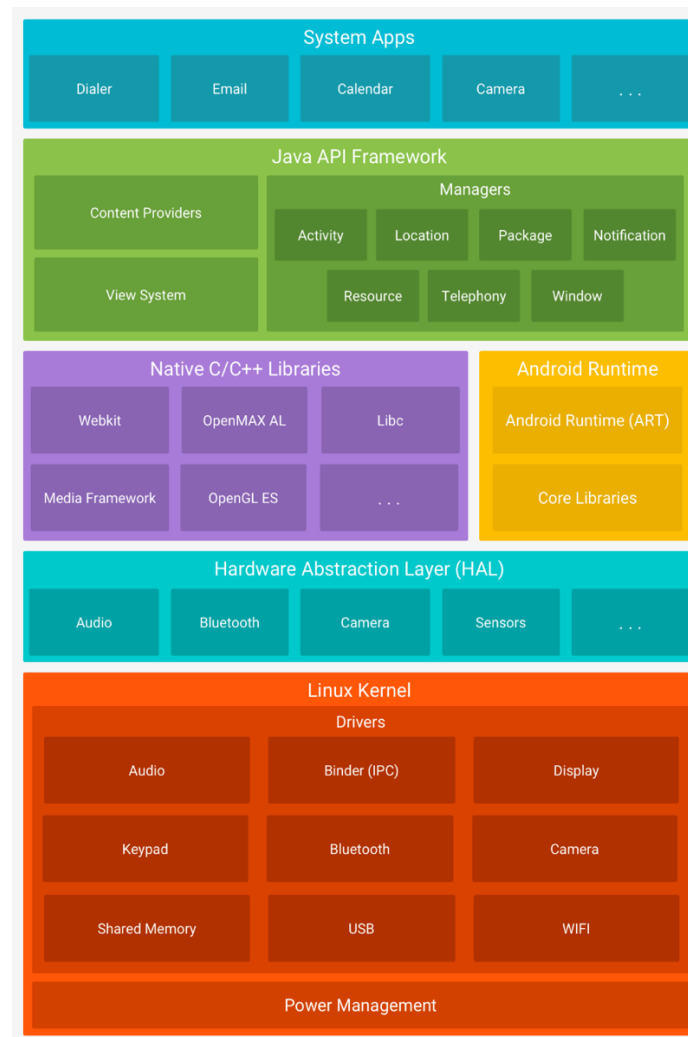
- Metadatos: una lista de acciones para verificar que la nueva versión es de aplicación en el slot objetivo.
- Datos adicionales: normalmente, el binario que contiene la actualización del sistema.

315.El slot activo se marca como válido para el arranque a través de "*markBootSuccessful*()".

- 316.El slot inactivo (slot objetivo o *target slot*) se marca como inválido para el arranque a través de "*setSlotAsUnbootable()*".
- 317.Las acciones definidas en los metadatos del payload se descargan en memoria y se ejecutan (la memoria utilizada se invalidará al finalizar la actualización).
- 318.Las particiones asociadas al slot objetivo se leen y se verifican sus hashes.
- 319.Se lanza el proceso de post-instalación: en caso de error, se reintenta la actualización. Si termina con éxito, el slot objetivo se marca como activo mediante "*setActiveBootSlot()*".
- 320.Cuando tenga lugar el siguiente reinicio del dispositivo móvil, se intentará arrancar del nuevo slot activo (que corresponde al que ha sido actualizado) siguiendo un esquema similar al de la imagen anterior. En caso de arranque exitoso, el slot se marcará como válido para el arranque a través de "*markBootSuccessful()*".
- 321.Después del arranque, y antes de que se lance el demonio zygote, se comprobará la integridad de las particiones a través de *dm-verity* (ver apartado "7.9. Proceso de arranque seguro: Verified Boot") antes de que pueda producirse cualquier cambio que impida una vuelta atrás.
- 322.En Android 7.x, el arranque posterior a la instalación de una actualización ya no lleva asociado el proceso de optimización de aplicaciones de versiones anteriores, aunque este proceso no tiene que ver con el mecanismo de actualización sin interrupciones, sino con el nuevo proceso de pre
- 323.compilación de ART (también introducido por Android 7.x) conocido como AOT (Ahead of Time).

7. MODELO Y ARQUITECTURA DE SEGURIDAD DE ANDROID

- 324.El modelo o arquitectura de seguridad de Android está basado en el sistema operativo Linux y proporciona adicionalmente numerosas librerías y entornos de ejecución para las aplicaciones móviles [Ref.- 11]:



325. Android intenta aplicar el principio de mínimos privilegios por defecto, donde cada aplicación móvil (o app) únicamente dispone de acceso a sus propios componentes, incrementando así la seguridad general de la plataforma, y limitando las acciones que una app puede realizar sobre otras apps, en el sistema o en su interacción con el usuario.

326. Para mejorar la seguridad de los ficheros privados, Android 7.x establece un acceso de tipo 0700 en Linux (lectura y escritura solo para el propietario) al directorio privado bajo "/data/data/" de cada app basada en la API 24. Esta restricción previene el acceso a metadatos de los ficheros privados de la app, como el tamaño o la propia existencia de un fichero.

327. Como consecuencia de lo anterior, el DownloadManager (gestor de descargas) deja de poder compartir ficheros almacenados de forma privada.

328. Las apps no pueden exponer recursos de tipo "file://" fuera de su sandbox, y, para compartir ficheros con otras apps, deben ofrecerlos a través de un proveedor de contenidos, permitiendo el acceso temporal al recurso.

7.1. MODELO DE PERMISOS DE ANDROID

329. La seguridad más granular de la plataforma Android se basa en un mecanismo de permisos que impone restricciones en las operaciones que un

proceso (o app) puede llevar a cabo, incluyendo también restricciones en el acceso a datos (en función de su identificador o URI, *Uniform Resource Identifier*) para los proveedores de contenidos (descritos posteriormente).

330. Se distinguen dos modelos de permisos: en tiempo de instalación y en tiempo de ejecución, que se detallan a continuación.

7.1.1. MODELO DE PERMISOS EN TIEMPO DE INSTALACIÓN

331. El presente apartado describe el modelo de permisos en tiempo de instalación de versiones previas de Android (hasta Android 5.x inclusive), y de aquellas apps que no tienen como versión objetivo Android 6.x o el nivel de API 23, versión en la que se introdujo el nuevo modelo de permisos en tiempo de ejecución. Es decir, las apps que se han sido desarrolladas con el modelo de permisos existente hasta Android 5.x, independientemente de la versión de Android sobre la que ejecuten, harán uso de este modelo.

332. Las apps por tanto deben definir y solicitar permiso para acceder a los recursos y datos compartidos en los que están interesadas. El sistema notifica al usuario estos permisos en el momento de la instalación de la app. Una vez el usuario autoriza a una app a disponer de dichos permisos, ésta será instalada y podrá hacer uso de los mismos en cualquier momento durante su ejecución (presente y futuras). En ningún caso se solicitará autorización al usuario durante la ejecución de la app, sino únicamente en el momento de su instalación.

333. Android no proporciona granularidad suficiente para seleccionar únicamente un subconjunto de todos los permisos solicitados por una app y restringir así su funcionalidad. Si se instala una app, ésta dispondrá de acceso a todos los permisos solicitados. Si el usuario no está interesado en proporcionar ciertos permisos a una app, la única opción disponible es no instalar dicha app.

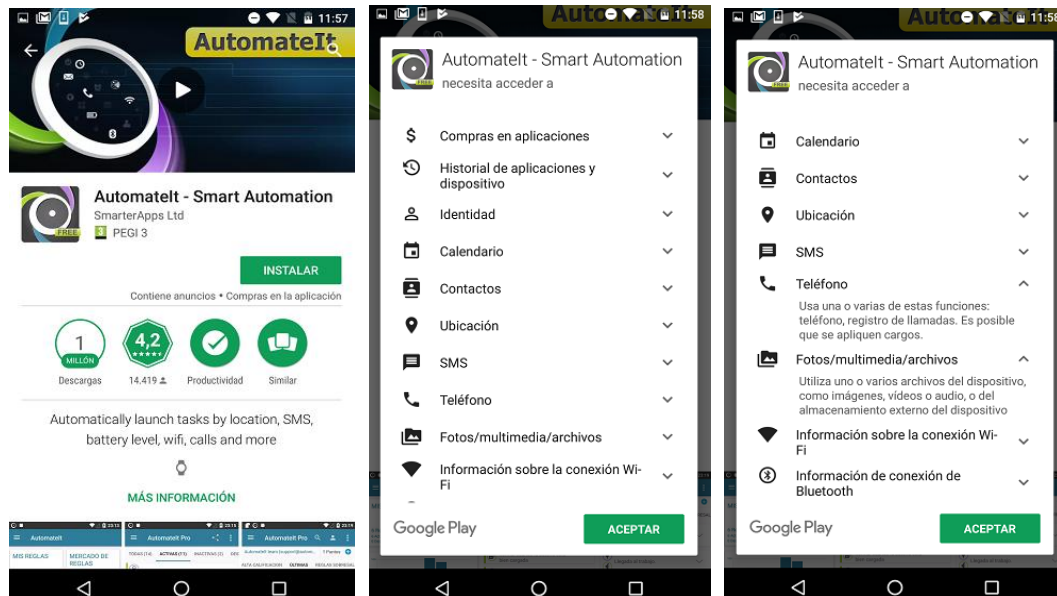
334. Adicionalmente, para ciertos permisos, como el de acceso a Internet, sería deseable disponer de suficiente granularidad en Android para poder autorizar a una app a conectarse a Internet sólo a ciertos sitios web, y no a cualquier destino. Esta funcionalidad no está disponible en el modelo de permisos actual de Android, y las apps con el permiso de acceso a Internet pueden conectarse a cualquier dominio o dirección IP.

335. La única excepción al modelo de permisos general descrito es la asociada a ciertos permisos disponibles para los desarrolladores (como por ejemplo "READ_LOGS", "WRITE_SECURE_SETTINGS", etc.), introducidos en Android 4.2 y que si podían ser habilitados o deshabilitados bajo demanda en cualquier momento y de manera dinámica, mediante el comando "pm grant/revoke", *Package Manager*.

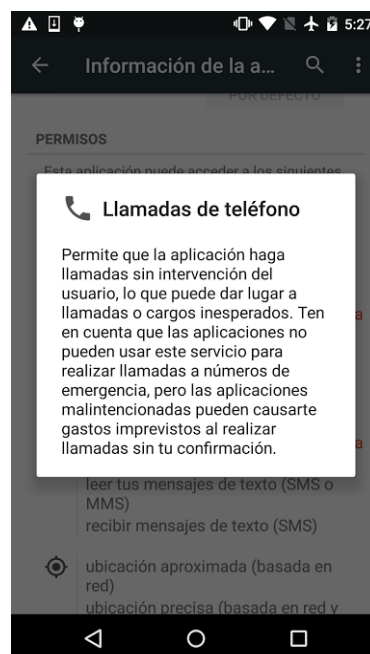
336. El siguiente ejemplo³⁸ muestra los diferentes permisos solicitados durante el proceso de instalación por una app Android concreta, que, aunque ejecutará en la versión 6.0.1 del sistema operativo, no ha sido desarrollada para el

³⁸ Se ha empleado la app "Automatelt" como ejemplo debido al elevado número de permisos que solicita al usuario.

nivel 23 de la API. En la aplicación móvil para la descarga de apps de Android, "Play Store", tras pulsar el botón de instalación en la página de descripción de la app, se solicita la aceptación por parte del usuario de los permisos demandados por la app [Ref.- 623]:

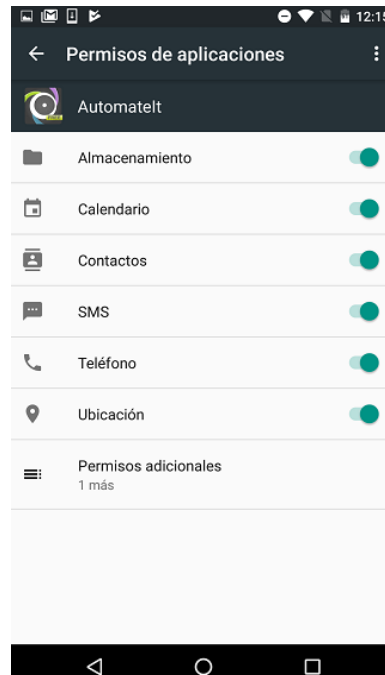


337. En versiones antiguas de la app "Play Store", al seleccionar de cualquiera de los permisos era posible obtener una descripción breve sobre su propósito (la imagen inferior ilustra la asociada al permiso de acceso al teléfono). Esta información adicional ya no está disponible en versiones más recientes (la captura pertenece a la versión de la app "Play Store" de Android 5.1.1):



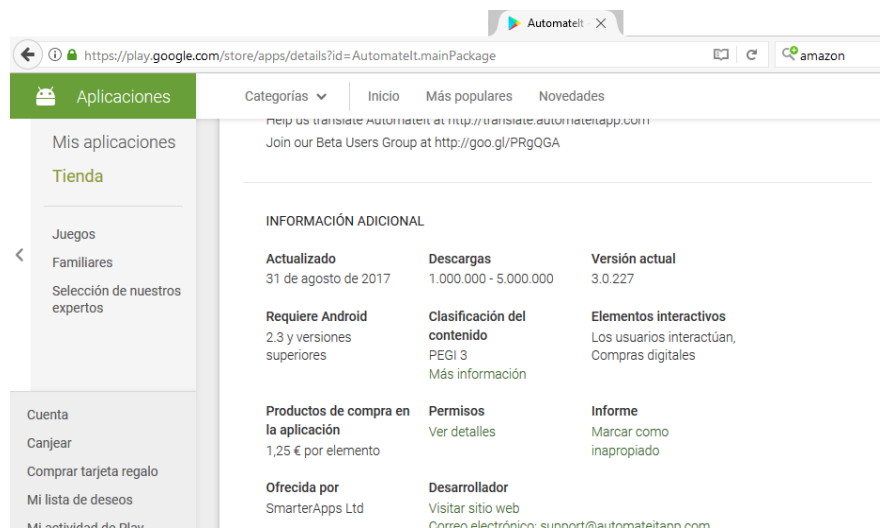
338. Una vez la app ha sido instalada, es posible ver los permisos de los que hace uso desde el menú "Ajustes [Dispositivo] - Aplicaciones". En Android 6.x desaparecen las listas "Descargadas, En Ejecución, o Todas" existentes en versiones anteriores y se muestra una única lista de aplicaciones. Al seleccionar una aplicación, se abre la ventana "Información de la aplicación"

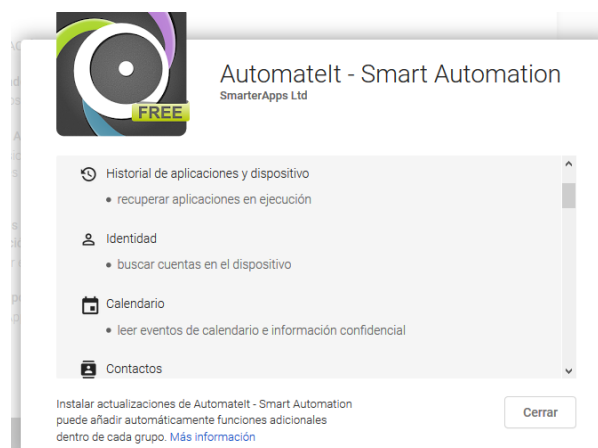
que incluye el campo "Permisos". Si se selecciona ese campo, se listarán todos los permisos en forma de interruptores que se pueden habilitar/deshabilitar individualmente:



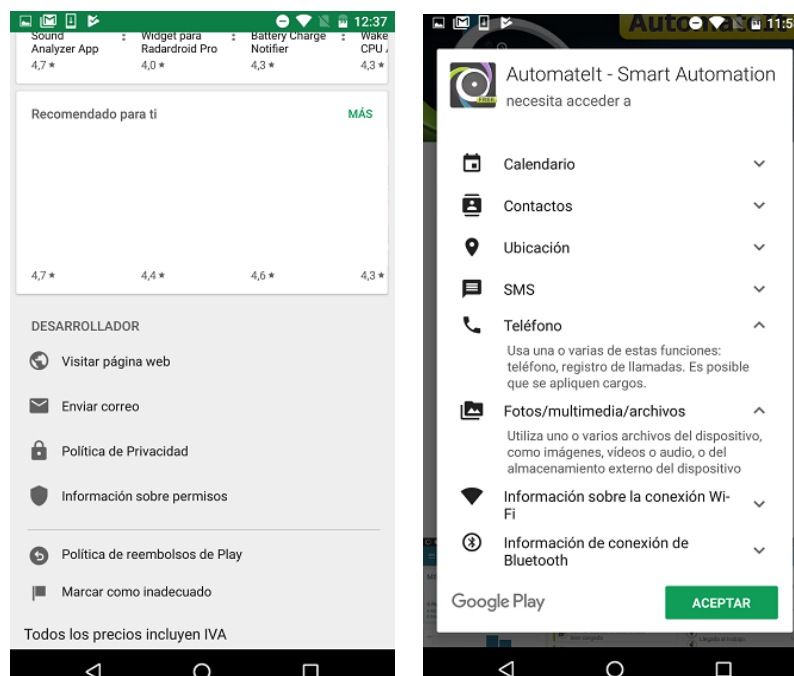
339. Por defecto, las apps no disponen de ningún permiso. Si durante la instalación una app no solicita ningún permiso no quiere decir que (debido a vulnerabilidades en Android) no pueda llevar a cabo acciones que afecten a la privacidad del usuario [Ref.- 55], por ejemplo mediante el envío de mensajes (denominados *intents* en Android) a otras apps, por lo que el usuario debe evaluar minuciosamente la instalación de nuevas apps incluso cuando éstas parecen no requerir ningún permiso.

340. Los permisos requeridos por una app también están disponibles a través de la página web de Google Play, junto a su descripción, que puede ser consultada antes de su instalación. Para ello, es necesario deslizar la barra de desplazamiento hasta la parte inferior y consultar el apartado "Permisos - Ver detalles":





341. También la app "(Google) Play Store" ofrece la posibilidad de consultar los permisos de la app bajando hasta el apartado "Desarrollador" y pulsando en la opción "Información sobre permisos":



342. Un elevado porcentaje del código dañino (o malware) para Android se distribuye utilizando el código de otras aplicaciones móviles legítimas, que son modificadas y publicadas de nuevo en el mercado oficial de aplicaciones, Google Play, o en mercados de terceros.

343. En otros casos, la app fraudulenta no incorpora ningún código dañino durante la instalación, y éste es añadido al dispositivo móvil posteriormente, en uno de los habituales procesos de actualización de las apps. Este proceso puede hacer uso de vulnerabilidades en Android que no requieren de aprobación por parte del usuario para la instalación de la nueva funcionalidad del código descargado, tal como ya demostró Jon Oberheide con "RootStrap" en el 2010 [Ref.- 225], ejecutando código nativo ARM fuera de la máquina virtual Dalvik.

344. Mediante técnicas de ingeniería social es también posible instar al usuario a realizar la actualización de una app maliciosa [Ref.- 126]. Se recomienda por

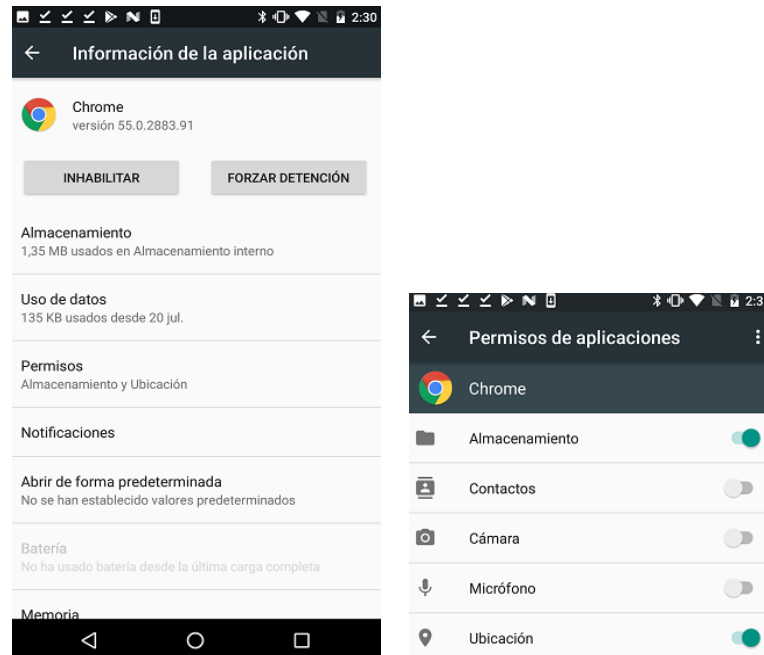
tanto ser muy precavido y evaluar la reputación y los permisos requeridos por el software antes de proceder a instalar cualquier app en el dispositivo móvil.

- 345. Por otro lado, debe tenerse en cuenta que si una actualización asociada a una app ya instalada requiere disponer de nuevos permisos no solicitados y aprobados por el usuario durante la instalación inicial de la app (o en versiones previas), el usuario será consultado de nuevo para aprobar los nuevos permisos requeridos por la actualización (ver apartado "20. Gestión de apps en Android").
- 346. Android 5.x estableció un nuevo requisito especificando y asegurando que sólo una app puede definir un permiso personalizado, a menos que esté firmada con el mismo certificado de desarrollador que la app que ya haya definido dicho permiso.
- 347. Cada app tiene la capacidad de definir cualesquiera permisos personalizados necesite. A partir de Android 5.x, si dos apps definen el mismo permiso personalizado, empleando el mismo nombre, o hacen uso de una librería que define dicho permiso, la instalación de la segunda app con el permiso duplicado se bloqueará. En versiones anteriores de Android, el sistema permitía la instalación de múltiples apps en estas condiciones, y otorgaba el nivel de protección asignado por la primera app que era instalada y que había definido el permiso personalizado.

7.1.2. MODELO DE PERMISOS EN TIEMPO DE EJECUCIÓN

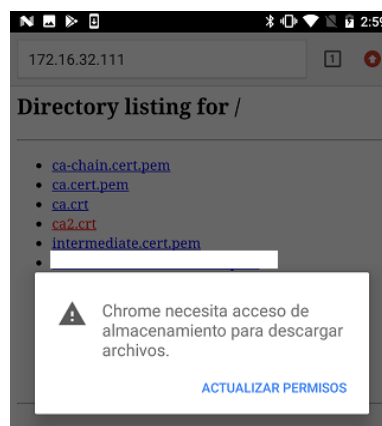
- 348. Android 6.x, vinculado a la API 23, introduce un nuevo modelo de permisos para las apps, donde los usuarios disponen de la capacidad de gestionar los permisos en tiempo de ejecución de las apps (*runtime permissions*) en lugar de en el momento de instalación de las apps, tal como ocurría en versiones previas de Android [Ref.- 617].
- 349. Este modelo proporciona un mayor control, visibilidad y granularidad por parte del usuario a la hora de conceder o denegar a las apps acceso a información y servicios. Los permisos pueden ser gestionados individualmente para cada una de las apps instaladas en el dispositivo móvil, y en cualquier momento.
- 350. Este nuevo modelo de permisos también proporciona al usuario un mejor contexto para comprender el propósito por el que una app solicita un permiso concreto.
- 351. Las apps que tienen una versión objetivo con un nivel de API igual (o superior) a 23 deben verificar si disponen del permiso antes de hacer uso del mismo, y en caso contrario, solicitarlo al usuario. Es decir, las apps que implementan la API 23 se instalan sin solicitar ningún permiso. Cuando se ejecutan por primera vez y requieren acceder a un recurso controlado por un permiso del sistema, solicitarán al usuario el permiso correspondiente. En este escenario, el usuario puede aceptar o denegar el permiso.

352. Tras la instalación de una app, la misma aparecerá sin ningún permiso de los considerados como "peligrosos" (ver apartado "7.1.3. Permisos normales y peligrosos") concedido:



353. Al igual que en Android 6.x, en Android 7.x los permisos disponibles en el menú anterior no son todos los permisos posibles considerados "peligrosos" por Google, sino aquellos que la app declara.

354. Existen algunas excepciones, como el caso del navegador web Chrome dónde, al arrancar la app o durante su funcionamiento, si el sistema detecta que Chrome está solicitando un permiso que aún no dispone, informará al usuario a través de un mensaje en el que se incorpora la opción "Actualizar permisos".

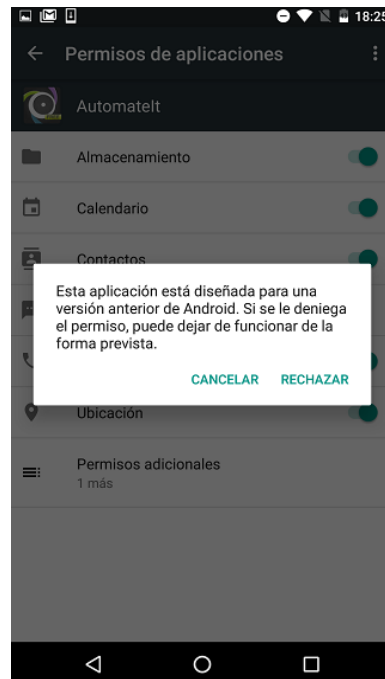


355. Al seleccionar dicha opción, el sistema irá presentando un mensaje para cada grupo de permisos requerido por la app de forma que el usuario otorgue y/o deniegue aquellos que considere oportuno. Esta secuencia de confirmación de permisos por parte del usuario es el comportamiento habitual para el resto de apps. En el ejemplo que se ilustra a continuación, se concede el permiso al almacenamiento compartido, tarjeta SD, pero se

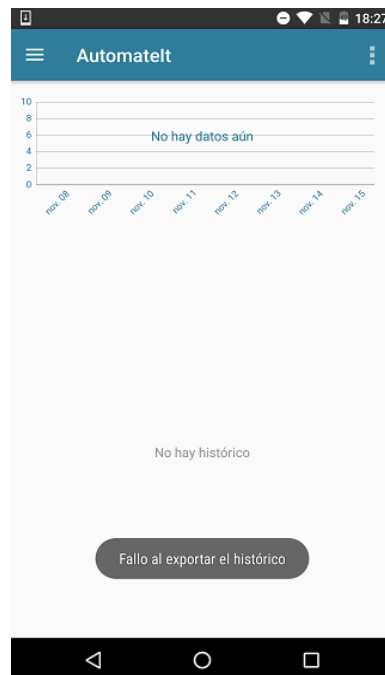
rechaza el acceso a los contactos. Esta circunstancia se refleja en los permisos mostrados en los ajustes:



- 356.A diferencia de lo que ocurre en versiones anteriores de Android (5.x o inferior), o con aplicaciones que ejecutan en Android 6.x pero implementan una API más antigua que la 23, si una app ve denegado su acceso a un permiso por parte del usuario, su ejecución no terminará necesariamente, sino que proseguirá, aun sin poder proporcionar la funcionalidad requerida, mostrando un mensaje informativo de la necesidad de otorgar el permiso para poder hacer uso de la funcionalidad en cuestión.
- 357.Algunas apps (como Mensajes o Messenger), pueden rehusar continuar su ejecución si no disponen de todos los permisos requeridos, si bien esta limitación no la impone el sistema, sino cada app individualmente.
- 358.Si algún grupo de permisos es denegado por el usuario, el sistema volverá a preguntar al usuario si desea concederlo la siguiente vez que se invoque la app. Si no se desea que este escenario se repita, se puede seleccionar la opción "No volver a preguntar" antes de rechazar el permiso en cuestión.
- 359.Si el usuario desea conceder un permiso ante una circunstancia especial pero no quiere que la app disponga de él permanentemente (por ejemplo, una aplicación de mensajería solicita acceso a la ubicación porque el usuario ha invocado una funcionalidad que la requiere puntualmente), ha de entrar en "Ajustes - [Dispositivo] Aplicaciones - [Nombre de la app] Permisos" y desmarcar los permisos que ya no se desea otorgar a la app.
- 360.Para las apps que, instalándose sobre Android 7.x, no implementan como mínimo la API 23, el sistema solicitará al usuario los permisos para la app en tiempo de instalación, siguiendo el modelo antiguo de permisos, y los mostrará en "Ajustes - [Dispositivo] Aplicaciones - [Nombre de la app] Permisos".
- 361.Desde dicho menú, el usuario podrá revocar los permisos que considere oportuno. El sistema mostrará un mensaje informando de las posibles consecuencias de esta acción al emplearse el antiguo modelo de permisos:



362. Si una app diseñada para APIs inferiores a la 23 solicita acceso a un recurso controlado por un permiso que el usuario ha revocado, el resultado será un fallo indeterminado o, incluso, una terminación anómala del proceso asociado a la app:



363. Por tanto, ante fallos de una app, se recomienda verificar si los permisos disponibles para ella son los requeridos y, en caso de que falte alguno, habilitarlo y ver si el problema se soluciona.

364. No obstante, no es infrecuente que algunas apps soliciten permisos que no precisan. Por tanto, en aras de la privacidad y la seguridad, se recomienda sopesar la necesidad de habilitar cada permiso y su implicación para el funcionamiento normal de la app.

365. Dentro de las mejores prácticas para los desarrolladores de apps, se recomienda consultar la documentación oficial de Google en relación a los permisos de sistema [Ref.- 618].

7.1.3. PERMISOS NORMALES Y PELIGROSOS

366. Adicionalmente, junto al nuevo modelo de permisos en tiempo de ejecución de Android 6.x, Google introdujo dos categorías de permisos: los permisos peligrosos (*dangerous permissions*) y los permisos estándar (*normal permissions*) [Ref.- 627].

367. Existe un permiso adicional, denominado "*signature*" (firma), que otorga el sistema si la aplicación que lo solicita está firmada con el mismo certificado que la app que declaró el permiso.

368. Se considera un permiso estándar por parte de Google aquel que, aunque requiera acceso a recursos ajenos al sandbox de la app, no compromete la privacidad del usuario (según Google). El sistema concederá ese permiso de forma automática si la aplicación lo declara en su fichero *manifest* y no consultará al usuario sobre ellos ni se los mostrará en la sección de permisos de la app. Un ejemplo de los permisos que Google cataloga como estándar es el acceso a Internet; otro es BLUETOOTH_ADMIN, que posibilita gestionar el interfaz Bluetooth.

369. Se considera un permiso peligroso (según Google) aquel que concede a la aplicación acceso a datos confidenciales del usuario, o que puede afectar al funcionamiento de otras apps. Los permisos así declarados en el fichero *manifest* de la app deben aprobarse explícitamente por el usuario. Un ejemplo es el acceso a los contactos.

370. Los permisos considerados peligrosos son: almacenamiento, calendario, contactos, cámara, SMS, teléfono, ubicación, micrófono y sensores.

371. Puesto que el usuario puede revocar los permisos de una app en cualquier momento, las apps deben chequear si disponen de ellos cada vez que intentan acceder a recursos protegidos por permisos.

7.1.4. PERMISOS SIMPLIFICADOS Y GRUPOS DE PERMISOS DE GOOGLE PLAY

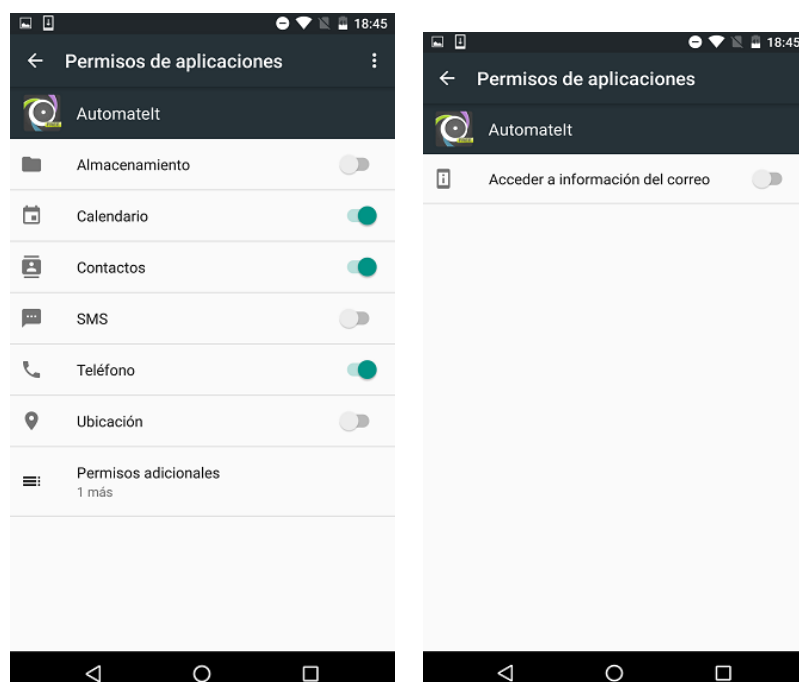
372. En relación al modelo de permisos de Android, Google introdujo una actualización mediante los Servicios de Google Play y la app "Play Store" en junio de 2014, en concreto en la versión 4.8.19³⁹ de "Play Store" (o "Google Play Store"), creando los permisos simplificados (o grupos de permisos) de Android.

373. Este cambio en el modelo de permisos de Android es de aplicación para todas las versiones de Android, desde la versión 4.4 de Android (por la fecha

³⁹ <http://www.androidpolice.com/2014/07/15/google-rolling-out-play-store-v4-8-19-with-paypal-support-simplified-app-permissions-bigger-buttons-and-more-apk-download/#simplified-permission-lists>

en la que fue introducido), afectando también a las nuevas versiones de Android.

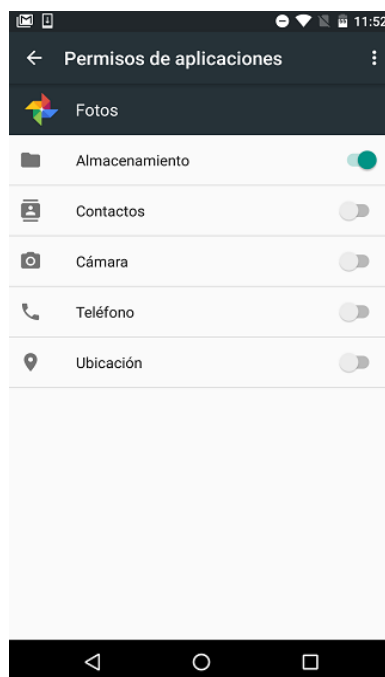
374. Los permisos individuales solicitados por cada app, en lugar de mostrarse a través de Google Play como en versiones previas, se han simplificado y agrupado en función de las capacidades o funciones específicas a las que está asociado cada uno de los permisos, añadiéndose un icono descriptivo asociado, y creándose grupos de permisos relacionados entre sí.
375. El objetivo de esta modificación fue simplificar el interfaz gráfico de usuario (GUI, *Graphical User Interface*) asociado al modelo de permisos de Android, de forma que los usuarios puedan entender más fácilmente qué tipo de capacidades son requeridas por una app en el momento de su instalación. Aproximadamente se han agrupado unos 150 permisos individuales en una docena de categorías o grupos de permisos.
376. Los grupos de permisos, a su vez, son ordenados según los criterios de Google por su criticidad o relevancia, mostrándose en la parte superior los más relevantes y disponiéndose de una categoría "Permisos adicionales" en la parte inferior, que engloba el resto de permisos solicitados y no agrupados previamente en los permisos principales [Ref.- 160]. Sin embargo, esta categoría adicional puede incluir también permisos relevantes desde el punto de vista de seguridad, como el acceso a los interfaces Bluetooth y NFC, el acceso al correo, o la capacidad de poder escribir en el historial y en los marcadores o favoritos web:



377. Asimismo, en el caso de las apps de terceros que definen sus propios permisos personalizados, estos también son englobados dentro de la categoría "Permisos adicionales".
378. Desde el interfaz de usuario, para las apps que no implementen al menos la API 23 (correspondiente a Android 6.x), es posible, durante el proceso de

instalación de la app, seleccionar los distintos grupos de permisos para ver los permisos individuales solicitados por la app.

379. Igualmente, se pueden consultar los permisos individuales antes de la instalación de una app, desde la app "Play Store" o desde la página web de Google Play, a través de la sección de "Desarrollador", seleccionando la opción "Información sobre permisos" disponible al final de la página de la app.
380. Asimismo, una vez se ha instalado una app, desde el menú "Ajustes [Dispositivo] - Aplicaciones" es también posible seleccionar la app y en la parte inferior, consultar los permisos individuales solicitados por la misma desde la sección "Permisos":



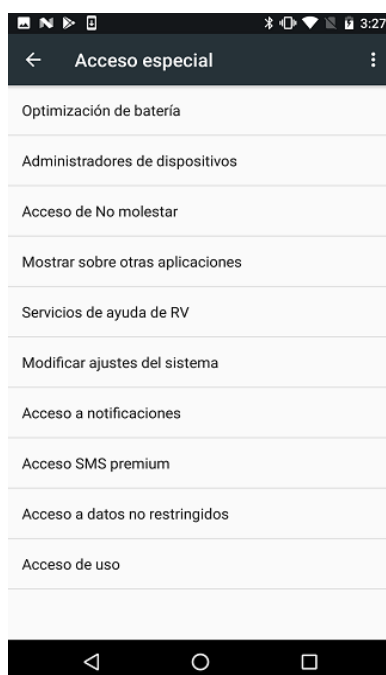
381. El problema radica en que la implementación de este nuevo modelo de permisos simplificados introduce nuevas vulnerabilidades de seguridad previamente inexistentes [Ref.- 161].
382. Por ejemplo, una app que requiera disponer del permiso para leer SMS, antes solicitaba únicamente el permiso "READ_SMS ", pero en las últimas versiones, aunque solicitará el mismo permiso, se le otorgará el grupo de permisos "SMS". Al ser instalada, la app obtiene permisos para acceder a todas las capacidades relacionadas con los SMS, incluyendo todos los permisos englobados en dicho grupo, como (potencialmente) el de envío de SMS, aunque el usuario no ha sido consultado sobre (y, por tanto, no ha aprobado) este permiso específicamente [Ref.- 162].
383. Por un lado, las apps que ya han sido instaladas y disponen de ciertos permisos, al recibir la siguiente actualización de la app, si ésta requiere de nuevos permisos, obtendrán automáticamente acceso al grupo completo de permisos asociados a cada uno de los permisos individuales de los que ya disponían. Como resultado, obtendrán automáticamente acceso a muchos más permisos de los que solicitaron inicialmente durante su instalación.

384. Por otro lado, los grupos de permisos contienen tanto permisos menos relevantes desde el punto de vista de seguridad, como permisos más críticos que son habitualmente utilizados por software malicioso. Por ejemplo, apps que solicitan el permiso de los servicios de localización o ubicación, podrán situar la posición del dispositivo móvil tanto de forma aproximada como más precisa a través del módulo GPS; apps que sólo requerían consultar el listado o registro de llamadas podrán realizar llamadas (incluso sin la intervención del usuario); y apps que solicitaban poder leer contenidos de la tarjeta SD podrán escribir también en ella o incluso formatearla.
385. Asimismo, por ejemplo, una app con funcionalidad de linterna (*flashlight*), que previamente requería del permiso para activar el *flash* (o led) de la cámara, al estar este permiso englobado dentro del grupo "Cámara", ahora también dispondrá de permisos para hacer fotografías, o grabar vídeo y audio.
386. Por último, esta actualización del modelo de permisos de Android otorga por defecto el permiso (existente previamente) de acceso completo a Internet a cualquier app. Es decir, el permiso de acceso a Internet ha sido eliminado de manera efectiva, ya que aunque los desarrolladores de apps tienen que seguir declarando en el fichero *manifest* que la app requiere dicho permiso, los usuarios no verán la solicitud de este permiso durante el proceso de instalación de la app (se puede mostrar en las vistas detalladas de permisos englobado dentro de la categoría "Otros"). Asimismo, las apps que no disponían de este permiso anteriormente, lo pueden obtener automáticamente y sin la aprobación del usuario en la siguiente actualización de la app.
387. Se recomienda por tanto analizar minuciosamente cada uno de los permisos individuales solicitados por las apps antes del proceso de instalación, durante el mismo, y al concluir éste, ignorando la simplificación en grupos de permisos realizada por Google.
388. Si una app solicita en su fichero *manifest* un permiso "peligroso" y no tiene aún ningún permiso perteneciente a ese grupo, el sistema pedirá al usuario confirmación para conceder acceso al grupo, no al permiso particular. Por ejemplo, si una app solicita el permiso `READ_CONTACTS`, el permiso solicitará acceso a los contactos.
389. Si una app solicita en su fichero *manifest* un permiso "peligroso" y ya dispone de algún permiso perteneciente a ese grupo, el sistema le otorgará el permiso automáticamente sin solicitar autorización al usuario. En el caso anterior, si una app obtuvo el permiso `READ_CONTACTS` y posteriormente requiere el permiso `WRITE_CONTACTS`, Android se lo concederá automáticamente.
390. Esta falta de granularidad supone un riesgo para la privacidad del usuario y la seguridad del dispositivo, pues una app que, en principio, solo requeriría acceso de lectura de la tarjeta SD, podría llegar a formatear la misma sin que el usuario pudiera evitarlo.

391. Adicionalmente, se recomienda llevar a cabo este mismo análisis minucioso para cada una de las actualizaciones recibidas para las apps ya instaladas en el dispositivo móvil, habiendo deshabilitado previamente las actualizaciones automáticas de las apps (ver apartado "20.2. Actualización automática de apps"), con el objetivo de evitar que una actualización de cualquier app adquiera nuevos permisos sin solicitárselos al usuario.
392. Por último, se debe asumir que actualmente cualquier app de Android dispone de capacidades plenas para conectarse y acceder a Internet.
393. Debe tenerse en cuenta que la simplificación del interfaz de usuario para la aprobación de los permisos requeridos por una app sólo afecta a las apps instaladas desde Google Play. La instalación de apps directamente mediante su fichero APK asociado todavía mostrará la lista individualizada de permisos, qué deberá ser aprobada por el usuario.
394. Se dispone de más información sobre los grupos de permisos, así como de los permisos individuales englobados en cada grupo, su propósito y capacidades, en la documentación oficial de Google [Ref.- 23].

7.1.5. PERMISOS DE ACCESO ESPECIAL

395. Android 7.x introduce un ajuste denominado "Acceso especial" dentro del menú "Ajustes - [Dispositivo] Aplicaciones - [icono de engranaje] Opciones avanzadas - Acceso especial".
396. Este menú incluye ajustes sensibles que permiten establecer excepciones sobre el comportamiento de Android, por lo que se recomienda revisar detalladamente las distintas categorías y comprobar qué apps se incluyen en la lista de excepciones.



397. Así por ejemplo, si no se desea que los servicios de Google Play ejecuten cuando no se está conectado a través de una red Wi-Fi para evitar consumo

indeseado de datos móviles, se recomienda desmarcar la entrada correspondiente a la app en la sección "Acceso a datos no restringidos".

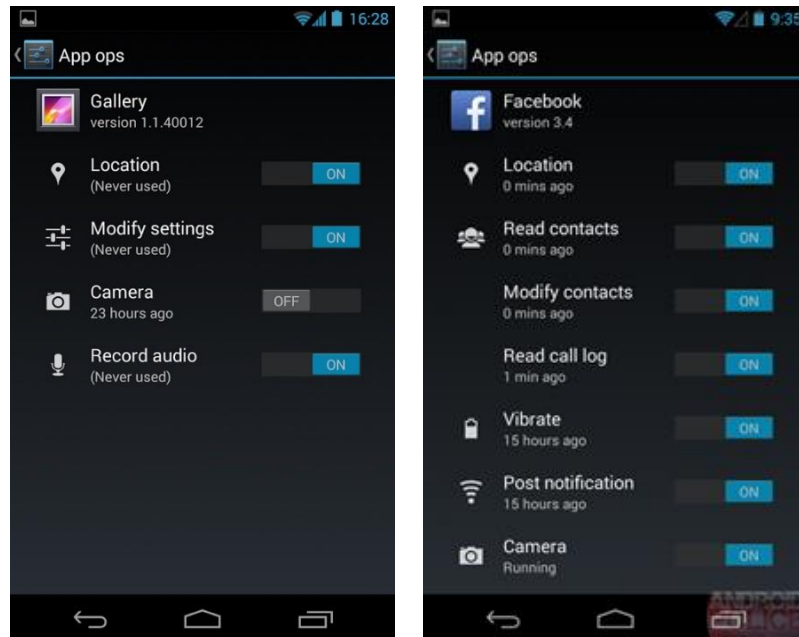
398. Es también conveniente revisar que no exista ninguna app dentro de la categoría "Administradores de dispositivos" que no sea el propio "Encontrar mi dispositivo" o el servicio "Google Pay" si este se encuentra configurado en el dispositivo móvil.
399. El permiso "Mostrar sobre otras aplicaciones" también se considera sensible, debido a los denominados ataques de superposición (ver apartado "20.5. Estadísticas detalladas de uso de las apps").
400. Por su parte, el permiso "Servicios de ayuda RV" corresponde a las configuraciones asociadas a visibilidad reducida (RV, Reduced Visibility).
401. De forma general, es importante verificar que no existe ninguna app dentro de cada categoría que no sea indispensable para la funcionalidad correspondiente.

7.2. APP OPS

402. "App Ops" es un sistema granular de control o gestión de los permisos de las apps en Android, oculto por defecto, y que fue introducido por Google en Android 4.3 [Ref.- 127], hasta la versión 4.4.2 [Ref.- 161].
403. La principal ventaja de "App Ops" es que podía ser usado para gestionar los permisos tras la instalación de cualquier app, en lugar de tener que aceptar o denegar los permisos en el momento de la instalación de la app, tal y como ocurre por defecto en el modelo de permisos original de Android. Esta funcionalidad permitía revocar permisos individualmente y para cada app.
404. Sin embargo, sólo es posible configurar cada permiso una vez que éste ha sido utilizado, en lugar de proporcionarse desde el inicio la lista completa de permisos de la app, por ejemplo, obtenida de su fichero "AndroidManifest.xml". Por otro lado, no se genera ninguna notificación hacia el usuario cuando un permiso solicitado por una app es denegado, por ejemplo, explicando el motivo de porqué no es posible hacer uso de dicho permiso.
405. Para poder hacer uso de "App Ops" era necesario crear un acceso directo personalizado que apuntase a la actividad asociada a "Ajustes - App Ops". La creación de este acceso directo se puede llevar a cabo a través de otra app creada específicamente para este propósito, como por ejemplo la app llamada "AppOps" [Ref.- 131] (entre otras⁴⁰).
406. "App Ops" proporcionaba controles granulares para permisos clasificados en diferentes categorías, como los servicios de localización, permisos personales, de mensajería o del dispositivo. Por ejemplo, permitía habilitar y deshabilitar la modificación de los ajustes, la cámara, la posibilidad de grabar audio del micrófono del dispositivo móvil, notificaciones, vibraciones del

⁴⁰ <https://play.google.com/store/apps/details?id=com.pixelmonster.AppOps&hl=en>

teléfono, lectura y modificación de los contactos, lectura del registro de llamadas, etc (imágenes correspondientes a Android 4.4.x):



407. "App Ops" permitía también ver la última vez que cada permiso había sido utilizado.
408. Posteriormente, en Android 4.4, "App Ops" seguía disponible y añadía capacidades para gestionar más permisos (como por ejemplo la capacidad de una app de despertar al dispositivo móvil, "keep awake") y una nueva categoría asociada a contenidos multimedia [Ref.- 128].
409. Sin embargo, finalmente "App Ops" fue eliminada por Google en la versión 4.4.2 de Android [Ref.- 161], confirmando su carácter experimental y el riesgo de que algunas apps dejaran de funcionar correctamente (ya que las apps esperaban disponer de los permisos solicitados una vez han sido instaladas y aprobadas por el usuario, en base al modelo inicial de permisos de Android).
410. Actualmente, sólo puede hacerse uso de "App Ops" en dispositivos móviles Android *rooteados* (por ejemplo, mediante un módulo Xposed denominado AppOpsXposed⁴¹).
411. En resumen, "App Ops" fue el precursor en Android 4.x del nuevo modelo de permisos en tiempo de ejecución que fue finalmente incorporado en Android 6.x (ver apartado "7.1.2. Modelo de Permisos en tiempo de ejecución").

7.3. SANDBOX DE EJECUCIÓN DE LAS APPS

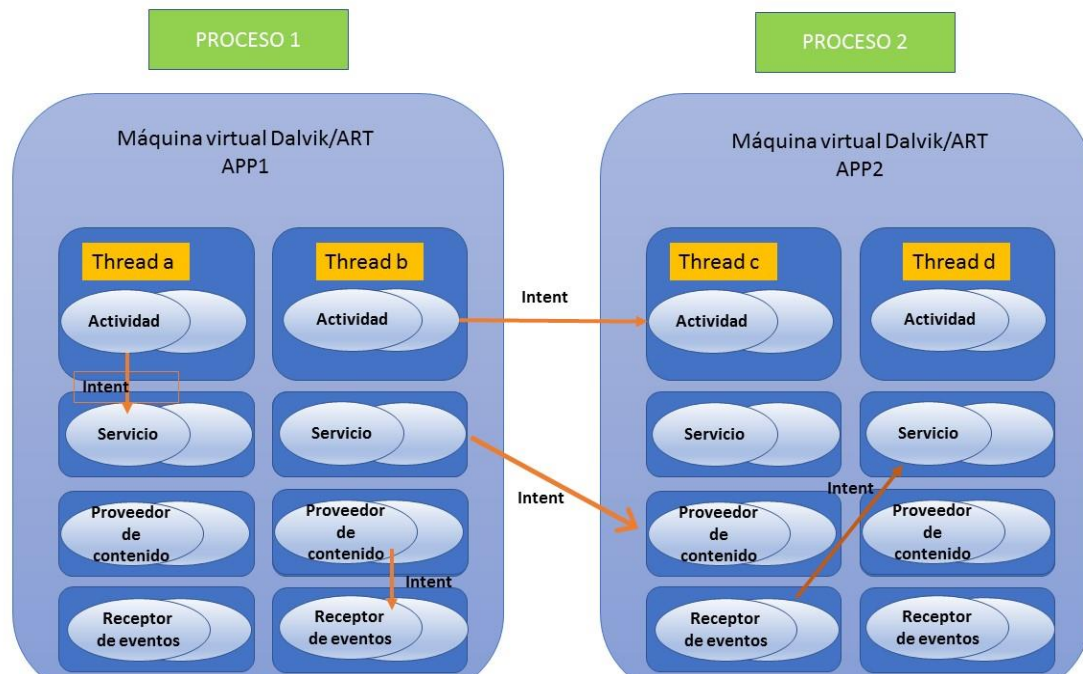
412. Las apps en Android ejecutan de forma independiente en su propio entorno de ejecución o *sandbox*⁴² [Ref.- 16]. Cada aplicación móvil (o app) está asociada a un usuario diferente del sistema operativo Linux.

⁴¹ <http://www.howtogeek.com/177915/how-to-restore-access-to-app-ops-in-android-4.4.2/>

413. En el momento de su instalación, Android asigna a cada app un identificador de usuario único (UID/GID, User ID y Group ID; ID en adelante) a nivel de sistema operativo, y asigna todos los ficheros de la app a dicho ID. De esta forma, sólo la app tendrá acceso a sus ficheros asociados.
414. El ID es asignado y gestionado por Android y las apps no tienen conocimiento del mismo.
415. En versiones anteriores de Android, una app podía excepcionalmente crear datos (directorios y ficheros) y asignarles permisos de lectura y/o escritura globales, de forma que cualquier otra app pueda acceder a ellos.
416. Con Android 7.x el directorio privado de las apps ("/data/data/") hace uso de los permisos 0700 de Linux para las apps cuya versión objetivo de Android es el nivel de API 24 (Android 7.x), o superior.
417. Desde el punto de vista de la compartición de ficheros entre apps, Android 7.x impide, a través de la política "StrictMode" [Ref.- 724], que los recursos de tipo fichero (file://) puedan ser accedidos desde fuera de la app; si una app quiere compartir datos con otras apps, deberá optar explícitamente a esta funcionalidad ofreciendo sus ficheros a través de un proveedor de contenidos (Content Provider), mediante un enlace "content://".
418. Cada app ejecutada es un proceso (Linux) independiente a nivel de sistema operativo. El proceso es iniciado cuando cualquiera de los componentes de la app necesita ejecutarse, y es finalizado al no utilizarse más la app o durante las tareas específicas de gestión (y liberación) de memoria del sistema operativo.
419. Cada proceso tiene asignada su propia máquina virtual (VM, *Virtual Machine*) Dalvik, por lo que el código de cada app ejecuta de manera aislada al resto de apps.
420. Las apps pueden compartir datos con otras apps a través de la compartición de un mismo ID (si han sido creadas por el mismo desarrollador), y acceder a servicios del sistema operativo mediante la solicitud de permisos para interactuar con otros componentes y datos del sistema, como la lista de contactos del usuario, los mensajes SMS, la tarjeta de memoria externa, la cámara, el interfaz Bluetooth, disponer de acceso a Internet, el interfaz Wi-Fi, poder realizar llamadas de voz, etc.
421. La lista de permisos disponibles en Android está disponible en la definición del fichero *manifest* (o manifiesto, descrito posteriormente) donde se especifican todos los permisos requeridos [Ref.- 23].
422. Las apps que comparten un mismo ID tienen acceso a los mismos ficheros, pueden incluso ejecutar en el mismo proceso y compartir la misma máquina virtual Dalvik. Para ello deben haber sido firmadas con el mismo certificado digital, asociado a su desarrollador, y desde el punto de vista de seguridad, podrían considerarse la misma app.

⁴² <https://source.android.com/devices/tech/security/index.html>

423. Las apps de Android pueden estar constituidas por diferentes componentes, que permiten la interacción de la app con el sistema y con el usuario, existiendo cuatro tipos de componentes: actividades (*activities*), servicios (*services*), proveedores de contenidos (*content providers*) y receptores de anuncios o eventos (*broadcast receivers*) [Ref.- 16].



424. Los mensajes asíncronos de notificación o solicitud de peticiones entre componentes (dentro de la misma o entre distintas apps) para que se lleve a cabo una acción se denominan *intents*.

425. Cada app de Android dispone de un fichero de definición, o *manifest* ("AndroidManifest.xml") [Ref.- 105][Ref.- 18], que contiene información sobre los diferentes componentes que constituyen la app. Adicionalmente, el *manifest* incluye los permisos que requiere la app (por ejemplo, acceso a Internet, acceso a la lista de contactos, etc.), las librerías (o APIs, *Application Programming Interface*) empleadas por la app, tanto de sistema como externas (por ejemplo, la librería de Google Maps), o los elementos o capacidades hardware y software empleados por la app (ej. cámara, interfaz Bluetooth, pantalla táctil, etc.).

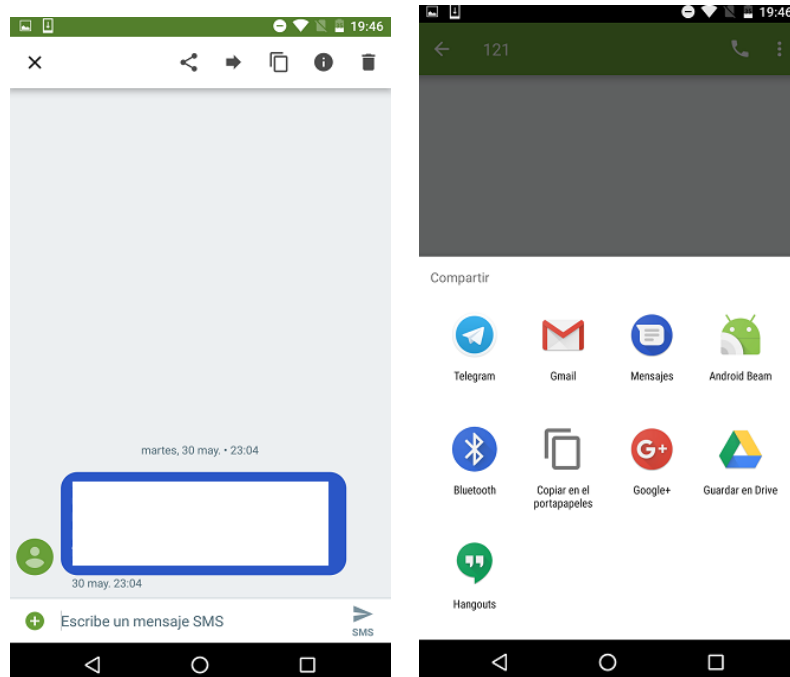
426. Estos requisitos hardware y software pueden ser empleados para decidir en qué dispositivos móviles puede instalarse y ejecutar una app en función de sus necesidades, principalmente desde Google Play. Los requisitos pueden ser definidos como necesarios o requeridos, u opcionales, en cuyo caso la app podrá ser instalada pero no hará uso de la funcionalidad asociada al requisito no disponible, como por ejemplo, la cámara.

427. Adicionalmente a los permisos estándar [Ref.- 23], existen una serie de permisos que requieren disponer de un nivel de protección elevado (*protectionLevel*) y que están disponibles únicamente para las apps existentes en la imagen de sistema de Android o que han sido firmadas con los mismos certificados que éstas. [Ref.- 105].

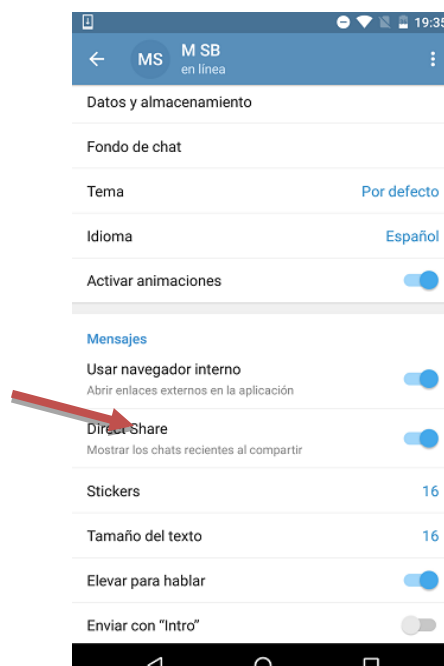
428. Las librerías de sistema, denominadas nivel de API (o API level) [Ref.- 19], e identificadas por su número o versión, están asociadas a la versión de Android. Por ejemplo, el nivel de API 24 corresponde a la versión 7.0 de Android ("Nougat").
429. Adicionalmente al código, las apps disponen de recursos (definidos mediante ficheros XML) asociados principalmente a la parte visual o interfaz gráfico de la app, es decir, imágenes, ficheros de audio, contenidos en diferentes idiomas, etc.
430. Con el objetivo de incrementar el modelo y los mecanismos de seguridad disponibles en Android por defecto, la NSA publicó en enero de 2012 una versión de Android restringida, denominada SE Android, y equivalente en filosofía y funcionalidad a SELinux para las plataformas Linux estándar [Ref.- 106].
431. Security Enhanced Android, o SE Android, mejora la seguridad de Android limitando el daño que puede infringir un código dañino, o una potencial vulnerabilidad de seguridad, que afecte al dispositivo móvil, así como añadiendo capacidades para aislar las apps, ficheros y otros componentes entre sí.
432. Posteriormente, las capacidades de SE Android fueron integradas en la plataforma móvil estándar, tal como se describe en el apartado "7.8. SELinux en Android".

7.4. COMPARTICIÓN DE DATOS ENTRE APPS Y DIRECT SHARE

433. A fin de simplificar la compartición de datos entre apps, incluso aunque no estén relacionadas, Android dispone de mecanismos por los cuales las apps pueden publicar sus capacidades para recibir datos de otras apps, que son expuestas al resto del sistema (y al usuario) a través del menú de compartición (o Share). Esta funcionalidad permite que otras apps puedan interactuar con los componentes de la app objetivo (o *target*) que publica sus capacidades, por ejemplo, lanzando una actividad y enviándole datos.
434. La app que quiere ser un objetivo con el que otras apps puedan compartir datos debe declarar el servicio de compartición asociado en su fichero *manifest*, solicitar el permiso `BIND_CHOOSER_TARGET_SERVICE` y crear un "<intent-filter>" con la acción `SERVICE_INTERFACE` [Ref.- 603].
435. Para hacer uso de la compartición de datos, al seleccionar un elemento susceptible de ser compartido, se mostrará la opción marcada con un grafo de tres puntos, el cual, al ser seleccionado, desplegará un menú con los iconos de las aplicaciones (o del contacto concreto dentro de esa aplicación) capaces de recibir ese elemento (o datos):

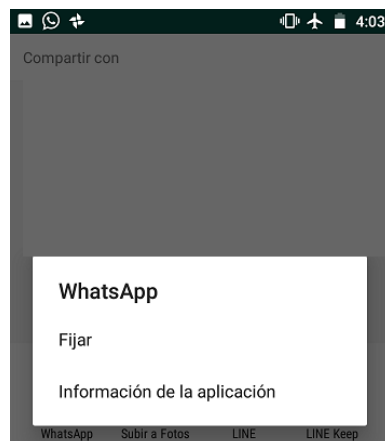


436. Complementariamente a la compartición, desde Android 6.x se incluye la funcionalidad "Direct Share", que permite que los datos compartidos se envíen a un submenú específico dentro de una app. Por ejemplo, se puede enviar un contenido (por ejemplo, una foto) a un contacto específico dentro de una app de mensajería, en lugar de a la app de manera general.
437. Debido a que cada vez más apps soportan esta característica, el menú de compartición puede aparecer muy cargado de elementos.
438. En Android 7.x no existe un ajuste que permita deshabilitar la funcionalidad "Direct Share" de forma global, quedando a discreción de cada aplicación el añadir en sus ajustes propios esta posibilidad, como ilustra la siguiente imagen:



439. Android 7.x permite ordenar la lista de apps disponibles para la función "Direct Share" a través del siguiente procedimiento:

- Seleccionar un elemento a compartir.
- Cuando aparezca la lista de objetivos "Direct Share", realizar una pulsación larga sobre la app que se desea que aparezca en primer lugar y seleccionar "Fijar".
- A partir de entonces, la app aparecerá en la sección superior del menú de compartición.



440. Si se han fijado varias apps dentro del menú de compartición, las mismas se mostrarán por orden alfabético.

441. Existen múltiples aplicaciones que implementan la posibilidad de configurar (incluyendo deshabilitar por completo) la función de compartición, pero todas ellas son apps de orígenes desconocidos, por lo que no se aconseja su instalación.

442. También es posible controlar el comportamiento de la función de compartición a través de un dispositivo móvil rooteado, lo cual está desaconsejado desde el punto de vista de seguridad.

443. De manera temporal, una opción para eliminar del menú de compartición determinados contactos o aplicaciones es eliminar el permiso de acceso a "contactos" de la app en cuestión.

444. Se dispone de un ejemplo de implementación de una app que emplea la característica "Direct Share" en la documentación oficial de Google para desarrolladores [Ref.- 17].

7.5. ART (ANDROID RUNTIME)

445. En versiones anteriores a Android 5.x (Lollipop), las apps para Android, que están desarrolladas en el lenguaje de programación Java, ejecutaban sobre una máquina virtual (VM, *Virtual Machine*) Java ligera denominada Dalvik [Ref.- 16].

446. La máquina virtual Dalvik se optimizó para su utilización en dispositivos móviles, con un consumo eficiente de memoria y de recursos. Dalvik ejecuta ficheros en formato Dalvik Executable (.dex), o DEX, por lo que en el proceso

de desarrollo el código fuente Java (archivos ".java", asociado habitualmente a archivos ".class", Java *bytecode*) de las apps de Android es convertido al formato DEX, Dalvik *bytecode* (mediante la herramienta "dx").

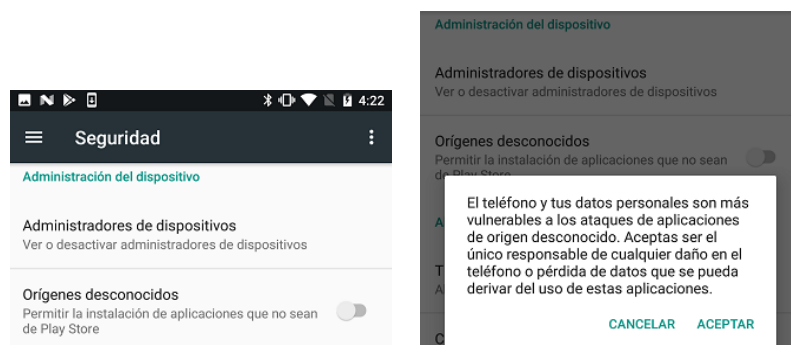
447. Android 4.4 (KitKat) introdujo el nuevo entorno de ejecución ART (*Android RunTime*) como sustituto de Dalvik [Ref.- 168], pero todavía de manera experimental y no habilitado por defecto. En Android 5.x, ART se convierte en el entorno de ejecución por defecto, sustituyendo a Dalvik.
448. ART no es compatible con binarios en el formato DEX original, y utiliza dos formatos de archivo: ".oat" y ".art" [Ref.- 196].
449. Algunas de las principales ventajas de ART frente a Dalvik es que hace uso de una técnica conocida como compilación AOT (*Ahead of Time*), que mejora el rendimiento de la ejecución de las apps, incorpora un proceso de recolección de memoria mejorado (GC, *Garbage Collector*), y otras mejoras de depuración [Ref.- 503].
450. La técnica AOT, además de aplicar optimizaciones en el uso del procesador (o CPU) y en la limpieza de procesos en memoria, precompila las aplicaciones para evitar que deba realizarse la traducción del *bytecode* DEX al ensamblador correspondiente al procesador del dispositivo móvil (ARM, Intel x86 o x64, etc.) en tiempo de ejecución. De ahí que los dispositivos móviles con Android 5.x requieran disponer de más espacio de almacenamiento asignado a los binarios optimizados de las apps, disponiendo por tanto de algo menos de espacio libre para los datos de usuario. Este proceso de precompilación de apps tiene lugar durante el primer arranque tras instalar o actualizar el sistema operativo (o posteriormente durante la instalación individual de las aplicaciones), y es por ello que se muestra el mensaje de optimización de aplicaciones cuando se actualiza la versión de Android.
451. Las herramientas de desarrollo de software de Android (Android SDK, *Software Development Kit*, o Android SDK Tools, y Android Studio⁴³ [Ref.- 22]) generan las apps para Android, o paquetes Android, en un único archivo o contenedor con extensión ".apk" (*Android PackAge*).
452. Adicionalmente al código Java, Android permite el desarrollo de aplicaciones nativas (ARM) usando C/C++, mediante el Android NDK (*Native Development Kit*), o híbridas, es decir, aquellas que contienen tanto código Java como código nativo (y hacen uso del JNI, *Java Native Interface*).
453. La mayoría de apps que ejecutaban anteriormente en Dalvik podrán ejecutar directamente en ART, salvo aquellas que hacen uso del JNI para ejecutar código nativo C/C++ (o ensamblador), hacen uso de técnicas de ofuscación, u otras técnicas de bajo nivel de limpieza de la memoria⁴⁴.

⁴³ <http://developer.android.com/sdk/index.html>

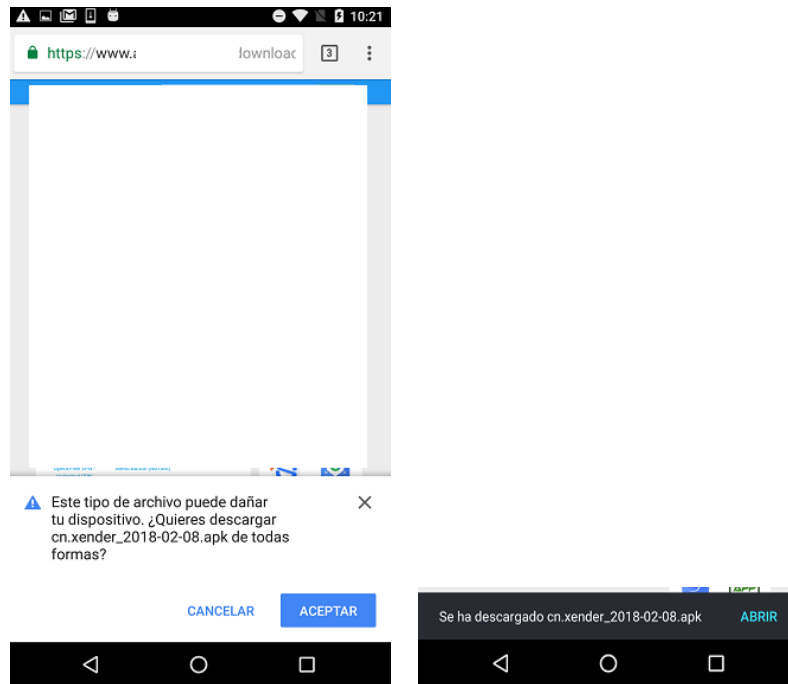
⁴⁴ <https://developer.android.com/guide/practices/verifying-apps-art.html>

7.6. INSTALACIÓN DE APPS (DE ORÍGENES DESCONOCIDOS) EN ANDROID

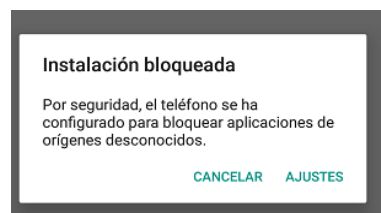
454. Se denomina aplicación móvil de "orígenes desconocidos" a una app que se instala en el dispositivo móvil empleando un canal diferente al oficial, es decir, al asociado a Google Play. Entre ellas, se incluyen las apps que se instalan a través de un fichero APK descargado directamente desde una página web o mensaje, que se cargan en el dispositivo a través de ADB(mediante comandos como "adb sideload" o "adb install"), o que se instalan a través de un mercado de apps de Android distinto al oficial.
455. Todas las apps para Android deben ser firmadas digitalmente por su desarrollador mediante un certificado digital, siendo por tanto posible identificar (teóricamente) al desarrollador de cualquier app oficial (ver apartado "7.7. Firma digital de apps en Android").
456. La firma de las apps es empleada adicionalmente por Android para establecer permisos de operaciones entre apps y distintos niveles de protección [Ref.- 105] (según el fichero *manifest*), y para permitir a las apps compartir su ID (si han sido programadas y firmadas por el mismo desarrollador).
457. El módulo estándar de instalación de apps de Android (install y Package Manager) verifica mediante certificados digitales que las apps oficiales, es decir, que provienen de Google Play, han sido firmadas convenientemente por su desarrollador.
458. Adicionalmente, es posible configurar el dispositivo móvil para permitir la instalación de apps no oficiales (no provenientes de Google Play) sin para ello ser necesario realizar el proceso de *rooting* sobre el dispositivo móvil.
459. Para permitir la instalación de cualquier app, incluso proveniente de fuentes que no son de confianza, simplemente es necesario eliminar las comprobaciones de seguridad asociadas en Android mediante la opción del menú "Ajustes [Personal] - Seguridad [Administración de dispositivos] - Orígenes desconocidos":



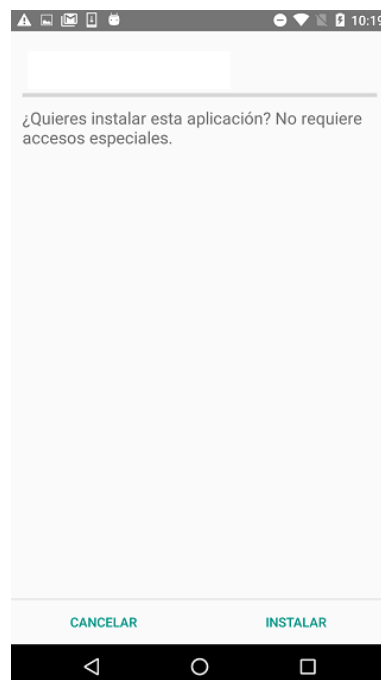
460. En ese estado, cuando se intenta descargar una app directamente desde una página web, Android mostrará un mensaje de advertencia al usuario:



461. En caso de que la opción "Orígenes desconocidos" no esté habilitada en el dispositivo móvil, se bloqueará su instalación:



462. Si la opción "Orígenes desconocidos" está activada, cuando se seleccione el nombre del fichero descargado (o se pulse en "Abrir") se mostrará un mensaje solicitando confirmación al usuario para instalar la app:



463. Desde el punto de vista de seguridad y con el objetivo de evitar la instalación de apps de fuentes u orígenes desconocidos, o que no son de confianza, se recomienda no habilitar esta opción, tal como advierte el mensaje mostrado por Android.

7.7. FIRMA DIGITAL DE APPS EN ANDROID

464. Para que una app sea considerada válida y pueda ser distribuida oficialmente a través de Google Play debe haber sido firmada digitalmente por su desarrollador [Ref.- 28]. Si una app no está firmada digitalmente, será rechazada por Google Play y por el instalador de paquetes del dispositivo Android.

465. Las apps de Android se distribuyen a través de ficheros APK (Android Package), una extensión del formato de paquetes Java JAR (Java ARchive). Por tanto, las apps hacen uso de capacidades similares a las existentes para firmar ficheros JAR para la firma de los paquetes o ficheros APK [Ref.- 44] [Ref.- 200].

466. La firma de una app establece la relación de confianza entre el desarrollador y Google en ambos sentidos: el desarrollador puede tener certeza de que su app se distribuye sin modificar (garantía de integridad), y a su vez es responsable del comportamiento de dicha app.

467. Cuando se instala el fichero APK perteneciente a una app, el Package Manager de Android comprobará que dicho APK está correctamente firmado con el certificado incluido en el APK. La clave pública del certificado identifica al desarrollador asociado con cada aplicación, haciendo posible junto al nombre identificador de la app que cada app ejecute con un id de usuario (UID) diferente y tenga asignado su propio sandbox, impidiendo que una app acceda a los datos de otra de forma directa (el acceso a datos de otra app solo es posible a través de mecanismos de comunicación entre procesos, IPC o Inter Process Communication).

468. Si se instala una nueva app y el Package Manager identifica que el certificado del APK se corresponde con el utilizado por otro APK previamente instalado en el dispositivo, se comprobará si el nuevo APK declara en su fichero manifest que quiere compartir el mismo UID que las apps ya instaladas y firmadas con ese mismo certificado, estableciendo así relaciones de confianza entre apps del mismo desarrollador y definiendo el nivel de privilegios de una app con respecto a las demás.

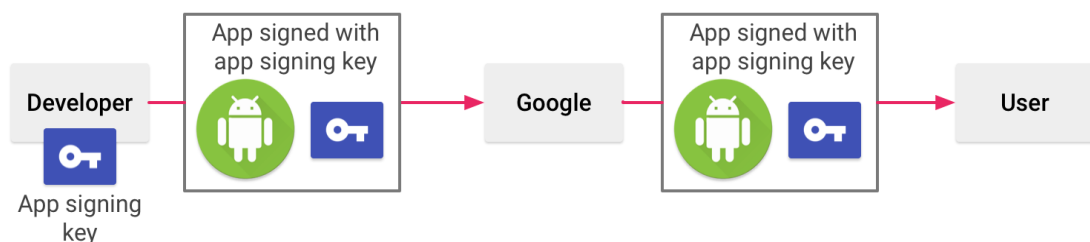
469. Cuando se actualiza una app, se verificará que la nueva versión de la app está firmada con el mismo certificado digital y que, por tanto, proviene del mismo desarrollador.

470. La firma digital permite también establecer permisos especiales basados en la firma de la app (signature permissions), de forma que las apps pueden declarar permisos a nivel de firma y limitar el acceso a estos permisos a apps firmadas con el mismo certificado, aunque tengan un UID diferente y, por tanto, diferentes sandboxes.

471. Android no realiza ninguna verificación adicional sobre la seguridad de la app o la firma digital (adicionalmente a las especificadas en el apartado "Verificación de seguridad de las apps: Google Play Protect" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]), salvo comprobar la fecha de expiración del certificado asociado a la firma en el momento de la instalación de la app⁴⁵, permitiéndose certificados auto-firmados y no generados o validados por ninguna autoridad certificadora (o CA).
472. Hasta Android 6.x, el modelo de firma de las apps se basaba en el firmado de los contenidos descomprimidos de los ficheros individuales contenidos en el APK (denominado "esquema de firmado de APK v1"). Android 7.x, junto a Android Studio 2.2, introdujo un nuevo esquema de firma (denominado "esquema de firmado de APK v2") basado en el firmado de los contenidos binarios del fichero APK completo [Ref.- 712].

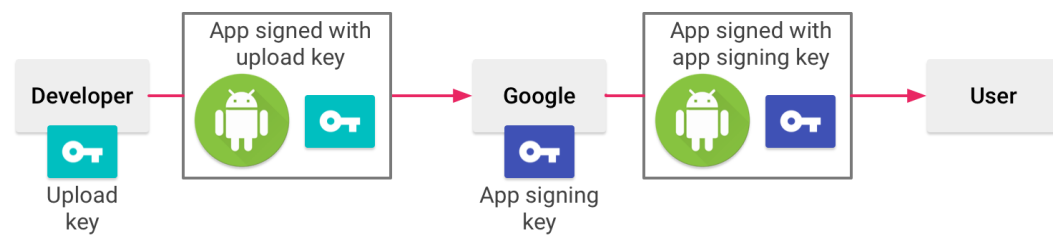
7.7.1. ESQUEMA DE FIRMADO DE APK V1 (JAR)

473. El modelo de firmado tradicional de firmado de apps se basa en una clave (denominada "app signing key" o "clave de firma de la app") a partir de la cual se genera un certificado digital que permite verificar la identidad del firmante del APK (el desarrollador). Para ello, el certificado digital utilizado para generar la firma se incluye en el propio fichero APK.
474. La firma se calcula sobre los contenidos descomprimidos de cada fichero del fichero APK, lo que añade mucha carga computacional al gestor de paquetes durante la verificación de la firma.
475. Dentro de este modelo, se ofrecen dos mecanismos de firmado:
- El desarrollador custodia su propio almacén de claves (keystore) y genera las firmas. La siguiente imagen ilustra este modelo, tradicional en Android.



- La gestión y custodia de la clave (o certificado) se delega a un keystore gestionado por Google [Ref.- 248]. Esta opción, que emplea la herramienta "Play Encrypt Private Key" proporcionada por Google Play, está desaconsejada desde el punto de vista de seguridad, pues implica que el desarrollador pierde parte del control sobre su clave (o certificado) de firma.

⁴⁵ Existen referencias confirmando la posibilidad de instalar apps incluso con certificados expirados [Ref.- 26].



476. Si se emplea el método de gestión de claves de Google, se requerirá, además de la clave (o certificado) de firma de la app, una clave que se conoce como "de subida" (o *upload*), la cual debe ser registrada en Google con anterioridad a la inclusión de la app en Google Play. Durante el proceso de subida o publicación de la app en Google Play, se utilizará la clave de subida para verificar la identidad del desarrollador de la app. La protección de esta clave de subida es responsabilidad del desarrollador, por lo que, si se ve comprometida, éste debe contactar con Google para revocarla.
477. Con este esquema, la firma digital de las apps en Android no evita que las mismas contengan vulnerabilidades de seguridad o código malicioso, sino únicamente permite vincular (teóricamente) la app a su desarrollador.
478. La vinculación entre la app y el desarrollador no ofrece fiabilidad desde el punto de vista de seguridad e identidad, ya que, al permitirse emplear certificados auto-firmados, los mismos pueden contener información que no está asociada al desarrollador real de la app. Es decir, la firma de una app no permite necesariamente identificar a su desarrollador real.
479. La firma de apps con el esquema de firmado v1 no protege todas las partes de la app (por ejemplo, no protege los metadatos del formato ZIP del fichero APK), y no es empleada (como sí puede ser el caso en aplicaciones Java) para la verificación de la integridad (es decir, la firma) de segmentos o porciones de código durante la ejecución de la app. Los controles en tiempo de ejecución de Android se basan en los UIDs/GIDs de los procesos de Linux asociados a la app [Ref.- 26].

7.7.2. ESQUEMA DE FIRMADO DE APK V2

480. Este esquema de firmado persigue simplificar la tarea del gestor de paquetes en la comprobación de firmas, reduciendo así los tiempos de instalación de las apps, y proporcionar mayor protección respecto a la integridad de los APKs.
481. Para ello, se inserta un nuevo "APK Signing Block" (bloque de firma criptográfico) dentro del fichero APK, inmediatamente antes del directorio central del formato ZIP. Dentro de este bloque se guarda una estructura por cada firmante, llamada "APK Signature Scheme v2 Block", que contiene [Ref.- 712]:
- Algoritmo de firma, digest, y firma: el digest se almacena para diferenciar la verificación de la firma de la comprobación de la integridad de los contenidos del fichero APK. Debe tenerse en cuenta que puede haber varios firmantes para un mismo fichero APK.

- Certificado digital del firmante (en formato X.509).
- Atributos adicionales en formato de pares "clave-valor".



482. Durante la instalación del fichero APK, la firma se computa sobre el contenido binario del fichero APK completo, en lugar de sobre los contenidos descomprimidos de los ficheros individuales, lo que garantiza la integridad del paquete en su conjunto.

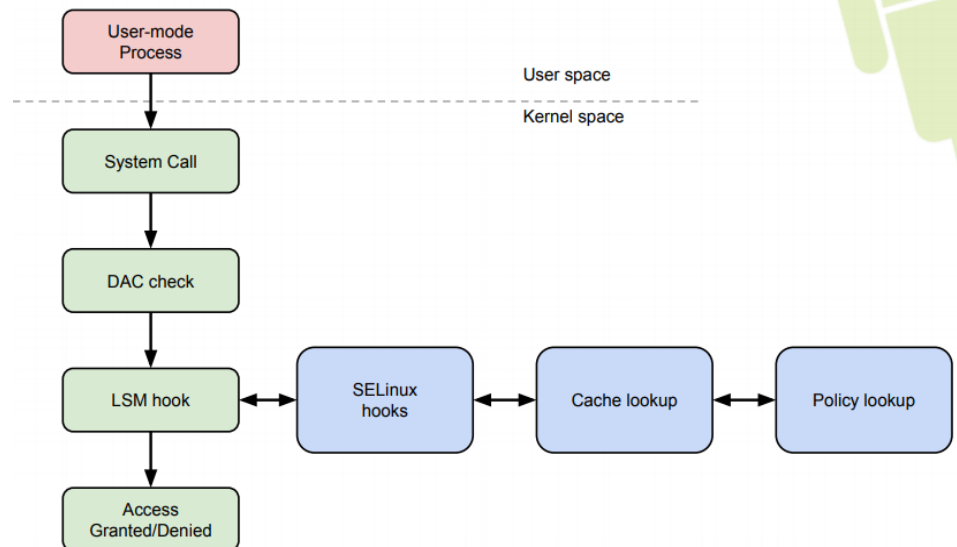
483. Un mismo fichero APK, es decir, una misma app, puede estar firmada mediante el esquema de firmado de APK v1 ó v2, para asegurar la compatibilidad hacia atrás con versiones previas de Android.

7.8. SELINUX EN ANDROID

484. El modelo de seguridad de Android se basa parcialmente en la utilización de contenedores (o *sandboxes*) para cada aplicación móvil (o app). La implementación previa a la versión 4.3 de Android empleaba un identificador de usuario (UID/GID) único de Linux, generado en el momento de la instalación de la app, para restringir el entorno de ejecución de cada app (ver apartado "7.3. Sandbox de ejecución de las apps"). Sin embargo, a partir de la versión 4.3 de Android, se emplea también SELinux (Security-Enhanced Linux) para restringir los límites y capacidades asociadas a una app y a su contenedor [Ref.- 106] [Ref.- 741].

485. SELinux proporciona soporte en Android para un modelo de seguridad conocido como MAC (*Mandatory Access Control*), frente al modelo de seguridad estándar y tradicional de Unix/Linux, conocido como DAC (*Discretionary Access Control*), junto a la definición asociada de políticas de seguridad. La siguiente imagen ilustra la arquitectura de SELinux [Ref.- 739]:

SELinux Architecture



486. Como ya se mencionó brevemente en el apartado "4.6. Android 4.3", las capacidades MAC de SELinux en Android se pueden emplear en todos los procesos, tanto los asociados a las apps como los procesos de sistema que ejecutan como root (u otros usuarios privilegiados), pero su implementación es dependiente de la versión de Android.

487. SELinux establece una serie de requisitos de seguridad que controlan los recursos accesibles por un proceso. Cada recurso (fichero, directorio, puerto, proceso, etc.) lleva asociada una etiqueta (o *label*), que también se denomina *contexto* en SELinux.

488. En Android 4.3 se hace uso de SELinux en modo permisivo o *permissive mode*, donde únicamente las acciones no legítimas son registradas por el kernel, pero no bloqueadas.

489. En Android 4.4 se hace uso de SELinux en modo obligatorio o *enforcing mode* parcialmente, es decir, sólo para un conjunto crítico de procesos o dominios de sistema (installd, netd, vold y zygote), en el que todas las acciones no legítimas son bloqueadas y registradas por el kernel. Para el resto de procesos o dominios, SELinux en Android 4.4 sigue actuando en modo permisivo, como en el caso de Android 4.3.

490. A partir de Android 5.0 (e incluyendo Android 7.x), se hace uso de SELinux en modo obligatorio o *enforcing mode* globalmente o completo (*full enforcement*), es decir, tanto para el conjunto crítico de procesos de sistema (ya protegidos en Android 4.4), como para el resto de procesos (con más de 60 dominios) [Ref.- 249].

491. En modo obligatorio, las acciones no legítimas son bloqueadas y registradas por el kernel en dmesg y logcat. Las entradas del subsistema de log de Android (logcat) pueden consultarse a través del comando "*adb logcat*". Las entradas que el kernel escribe directamente en sus logs pueden analizarse

con el comando "adb shell dmesg" o mediante el comando "adb shell su -c 'cat /proc/last_kmsg'".

492. Adicionalmente, ningún otro proceso puede ejecutar en el dominio de init (salvo el propio proceso inicial init).
493. Android 7.x introduce actualizaciones en la configuración de SELinux, incrementado la cobertura asociada a Seccomp [Ref.- 706].
494. Seccomp es una característica introducida en el kernel de Linux, disponible a partir de Android 5.x (Lollipop), que permite la implementación de sandboxes, es decir, entornos de ejecución restringidos donde un proceso, una vez ha sido restringido (o ejecutado de forma "segura"), sólo dispone de un número limitado de llamadas al sistema, y únicamente sobre los ficheros que tenía previamente abiertos [Ref.- 708]. Si el proceso intenta ejecutar alguna otra llamada al sistema, el kernel finalizará su ejecución.
495. Asimismo, Android 7.x introduce protecciones de memoria adicionales para las nuevas versiones del kernel de Linux, marcando nuevas zonas de memoria del kernel como de sólo lectura, y restringiendo el acceso del kernel a direcciones de memoria del espacio de usuario [Ref.- 722].
496. SELinux define etiquetas que se asocian a recursos, como sockets, ficheros y procesos, y que marcan qué acciones están permitidas [Ref.- 737]. Las políticas de acceso a un determinado objeto establecen qué se permite para una determinada etiqueta según este esquema:

```
allow dominio tipo:clase permisos
```

donde "dominio" es una etiqueta (o contexto) para el proceso o conjunto de procesos, el "tipo" es una etiqueta para el objeto al que aplica el acceso (por ejemplo, *fichero*, *socket*, *etc.*) , la "clase" es el tipo de objeto y el "permiso" es la operación a realizar (lectura o escritura).

Por ejemplo, la política "allow appdomain app_data_file:file rw_file_perms" otorga acceso de lectura y escritura a los ficheros etiquetados como "app_data_file" en el dominio o contexto "appdomain".

497. La implementación SELinux de Android incorpora elementos para proteger el espacio de kernel dividiendo los segmentos de memoria en secciones lógicas y estableciendo restricciones de acceso a su espacio de direcciones. Complementariamente, se restringe el acceso del kernel a la memoria de usuario (como se ha mencionado previamente).
498. Se restringe el acceso al sistema de medición de rendimiento del sistema (perf), de forma que en Android 7.x (Nougat) dicho acceso está bloqueado por defecto.
499. Se restringe el acceso de las apps a los comandos ioctl, de forma que sólo una pequeña lista de sockets ioctl está disponible para las apps [Ref.- 739], a fin de reducir la superficie de ataque limitando el acceso a identificadores de dispositivos persistentes, y evitar vulnerabilidades de escalada de privilegios

y ejecución de código en el contexto del kernel, como la descrita en [Ref.- 738].

7.9. PROCESO DE ARRANQUE SEGURO: VERIFIED BOOT

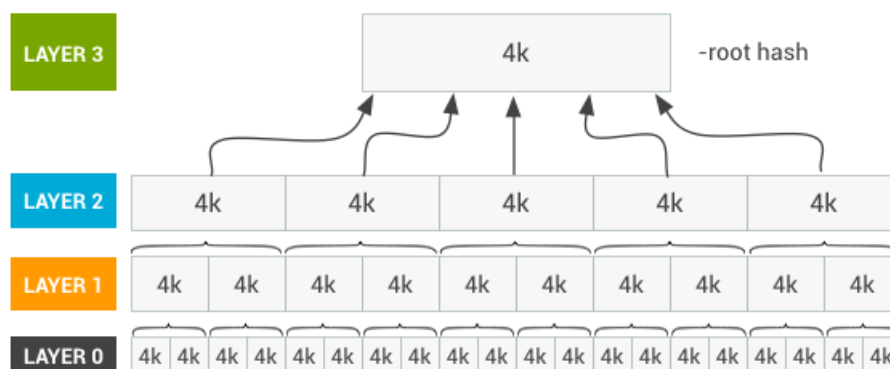
500.Verified Boot fue introducido en Android 4.4 ("KitKat") a través de las capacidades del kernel asociadas a device-mapper-verity (*dm-verity*) en dispositivos de almacenamiento de bloques (virtuales), con el objetivo de poder llevar a cabo verificaciones de integridad del proceso de arranque de Android, y evitar así su modificación, por ejemplo, por parte de software malicioso (malware) sin el conocimiento del usuario [Ref.- 34].

501.Verified Boot permite al usuario confirmar que el dispositivo móvil se encuentra en un estado conocido y verificado, es decir, con la partición de sistema ("/system") protegida y sin modificar.

502.Para ello, Android construye un árbol de hashes criptográficos (SHA256) asociados a los bloques (de 4 KB) de almacenamiento de la partición de sistema ("/system"), obteniéndose un hash raíz (*root hash*) que el fabricante del dispositivo móvil utiliza, junto a una clave de verificación propia (*verity_key*), empleada para verificar la integridad de todo el árbol (y de todos sus bloques asociados) y, por tanto, de toda la partición de sistema [Ref.- 512].

503.La clave de verificación se almacena en la partición de arranque (o "/boot"), cuya integridad a su vez también debería ser verificada por el código del gestor de arranque o *bootloader* (dependiente del fabricante del dispositivo móvil). Por tanto, desde el *bootloader* se verificará la integridad de los niveles siguientes que serán ejecutados: partición de arranque, partición de sistema, etc.

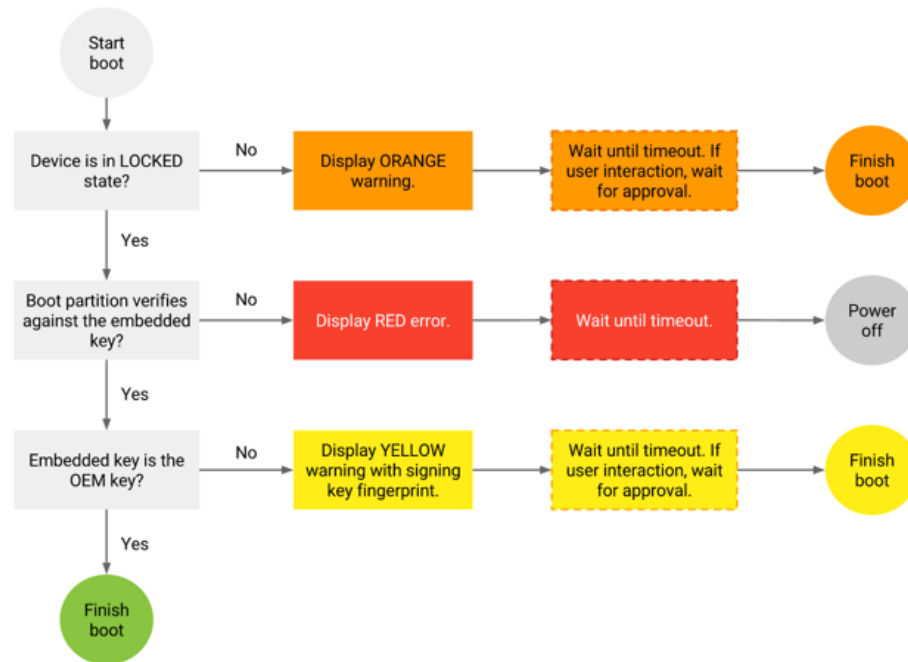
504.La modificación de un único bloque de la partición de sistema invalidaría el hash criptográfico raíz, y la tabla (o árbol) de hashes de integridad asociada, y permitiría detectar modificaciones en la integridad de dicha partición.



505. La detección de la integridad se realiza de manera automática por el kernel al leer datos de la partición de sistema, y verificar su contenido con la tabla de hashes de integridad. Cualquier modificación de la partición de sistema daría lugar a errores de entrada/salida (I/O o input/output), como sí ese bloque de datos no fuese legible.

506. Android 6.x añadió nuevas capacidades en Verified Boot para notificar al usuario durante el proceso de arranque acerca del estado e integridad del sistema [Ref.- 619].
507. Tras iniciar el dispositivo móvil, la acción `update_verifier` lanza la comprobación de integridad del sistema operativo a través de `dm-verity`, antes de que se arranque el demonio `zygote` para impedir que los servicios lleven a cabo cambios irreversibles, con el objeto de detectar modificaciones. Si la comprobación tiene éxito, se asume que el código a ejecutar viene de una fuente fiable (la versión oficial del sistema operativo) y `update_verifier` marca el arranque como exitoso. En caso contrario, se forzará un re arranque del dispositivo móvil.
508. Tras una actualización OTA, el proceso `update_verifier` leerá los bloques listados en `"/data/ota_package/care_map.txt"` (fichero proporcionado como parte del paquete OTA), aplicará los cambios de integridad sobre dichos bloques antes de reiniciar y actualizará el fichero con el árbol o tabla de integridad antes de reiniciar en la nueva versión.
509. La protección `dm-verity` está integrada en el kernel, y verifica los bloques individualmente según se van accediendo, en lugar de verificar directamente todos los bloques del sistema de ficheros durante el proceso de arranque del dispositivo, lo cual afectaría al rendimiento. Ante un fallo en la verificación, se generará un error que indica que el bloque no puede ser leído, lo que se traduce como que el sistema de ficheros está corrupto [Ref.- 727].
510. Aquellos bloques que han sido verificados se guardarán en caché para evitar tener que efectuar la verificación de nuevo en cada acceso al bloque. Este esquema de verificación parte del mecanismo "Verified Boot" desarrollado originalmente en el proyecto Chromium (los detalles de implementación están disponibles en la referencia [Ref.- 728]).
511. Si no es posible verificar la partición de sistema durante el proceso de arranque se pueden obtener tres mensajes de aviso (o estados del proceso de arranque), identificados por los colores naranja, rojo, amarillo o verde (arranque correcto)⁴⁶:

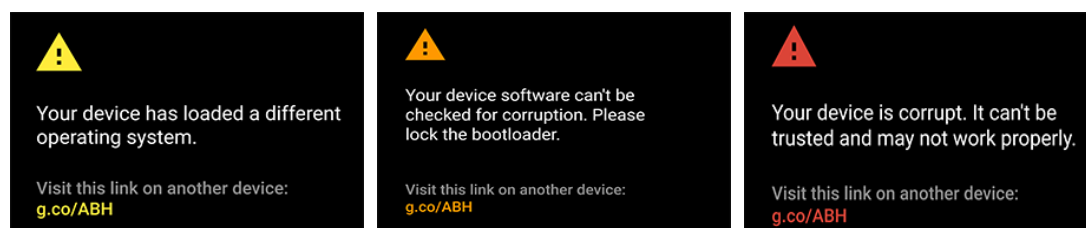
⁴⁶ <http://www.xda-developers.com/pixel-phones-not-yet-rootable-with-current-methods/>



512. "Your device has loaded a different operating system.", mostrado en color amarillo durante 10 segundos, indicando que se está arrancando desde un sistema operativo diferente al que venía de fábrica en el dispositivo móvil.

513. "Your device software can't be checked for corruption. Please lock the bootloader.", mostrado en color naranja durante 10 segundos, indicando que el gestor de arranque está desbloqueado, por lo que el dispositivo móvil permitiría arrancar desde imágenes de arranque (*bootloader*) o recuperación (*recovery*) diferentes y, como consecuencia, no es posible verificar la integridad del sistema operativo.

514. "Your device is corrupt. It can't be trusted and may not work properly.", mostrado en color rojo durante 10 segundos, indicando que el sistema operativo ha sido modificado y no es seguro hacer uso del mismo.



515. En el caso de haber desbloqueado el gestor de arranque intencionadamente, tras completar los cambios deseados, se recomienda volver a bloquear el gestor de arranque para evitar futuras modificaciones por parte de un tercero sin autorización (eliminando así el mensaje de aviso de color naranja).

516. Si el usuario identifica cualquiera de estos mensajes durante el proceso de arranque del dispositivo móvil, y no es consciente de haber llevado a cabo ninguna modificación de bajo nivel sobre el sistema intencionadamente, debería sospechar que el dispositivo móvil haya podido ser manipulado y/o comprometido.

517. Complementariamente, y directamente relacionado con Verify Boot, numerosos dispositivos móviles Android basados en procesadores Qualcomm pueden hacer uso del modo de arranque seguro conocido como Secure Boot [Ref.- 748].

7.10. MODO DE ARRANQUE SEGURO: SAFE MODE

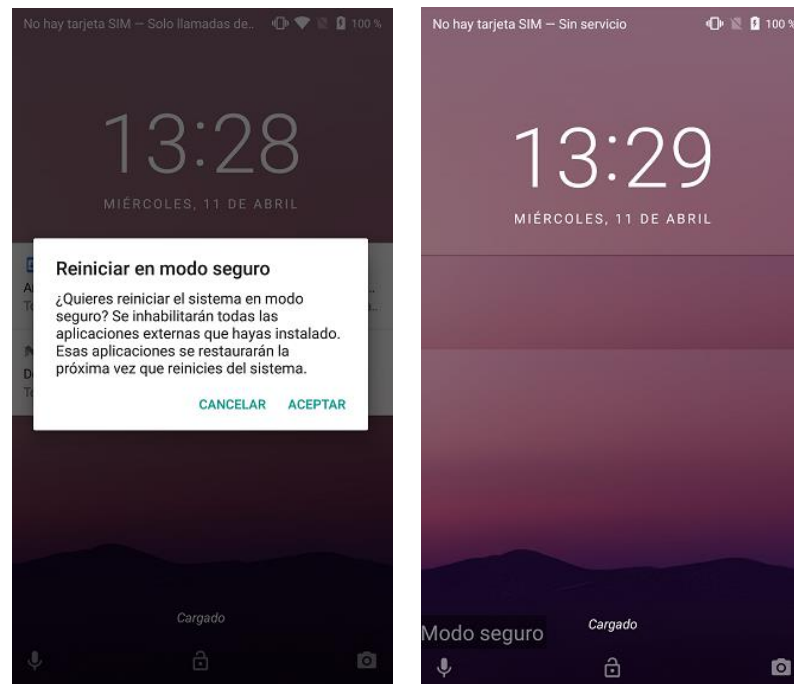
518. Además del modo "Verified Boot" descrito previamente (ver apartado "7.9. Proceso de arranque seguro: Verified Boot"), Android dispone de un modo especial de arranque que el usuario puede utilizar voluntariamente si detecta que el sistema presenta alguna inestabilidad, muestra un comportamiento extraño (como que se agote la batería rápidamente, que el dispositivo móvil se caliente en exceso, se quede sin memoria RAM disponible, lleve a cabo la ejecución de apps de manera ralentizada, etc.), o se sospecha que se ha podido instalar alguna app no fiable o inestable.

519. Se recomienda arrancar desde este modo de arranque si se sospecha que el dispositivo móvil se ha podido ver infectado con algún software o app móvil maliciosa (malware) [Ref.- 513].

520. Al arrancar en modo seguro o Safe Mode, Android ejecutará únicamente las aplicaciones preinstaladas, propias del sistema (es decir, disponibles en la partición de sistema o "/system") y los servicios básicos necesarios para el buen funcionamiento del sistema operativo Android.

521. El procedimiento para arrancar en modo seguro si el dispositivo móvil está encendido es el siguiente:

- Pulsar el botón de encendido (o apagado) hasta que aparezca el menú de apagado del dispositivo móvil.
- Seleccionar la opción "Apagar" y mantenerla pulsada hasta que el dispositivo informe al usuario de que va a reiniciarse en modo seguro.
- Confirmar la operación mediante el botón "Aceptar".



522. También es posible arrancar en modo seguro, cuando el dispositivo móvil está apagado, manteniendo pulsado el botón de bajar el volumen una vez aparece el logotipo de Google y se inicia su animación en la pantalla de arranque o inicio [Ref.- 513].
523. Cuando el sistema rearranque, mostrará en la pantalla de inicio el mensaje "Modo seguro" en el margen inferior izquierdo.
524. Todas las aplicaciones que no sean de sistema se mostrarán con sus iconos sombreados, siendo imposible ejecutarlas, pero sí desinstalarlas si se sospecha que alguna de ellas pueda estar ocasionando problemas en el dispositivo móvil.
525. Para salir del modo seguro, simplemente hay que apagar el dispositivo móvil (siendo en ocasiones necesario, según el modelo de dispositivo móvil, mantener el botón de apagado pulsado durante 30 segundos) y volverlo a arrancar con el botón de encendido de manera estándar.

Nota: Este procedimiento para arrancar en modo seguro (Safe Mode) es válido para dispositivos móviles Nexus, como el empleado en la elaboración de la presente guía. Para otros fabricantes, la combinación de teclas podría ser diferente.

7.11. ANDROID WEBVIEWS

526. Una `WebView`⁴⁷ es un componente del sistema basado en tecnología Chrome que permite a las apps mostrar contenidos web, como si se tratase de un navegador web personalizado, embebido y simplificado. Este contenido web puede incluir tanto páginas HTML como código JavaScript, que puede ser ejecutado por la app, por lo que las `WebViews` constituyen un componente crítico desde el punto de vista de seguridad en Android [Ref.-129].
527. Por ejemplo, mediante ataques de Cross-Site Scripting (XSS) e inyección de código JavaScript (por ejemplo, en ataques MitM, Man-in-the-Middle, especialmente cuando no se hace uso de técnicas de cifrado del tráfico mediante HTTPS ni de una correcta y estricta verificación de los certificados digitales), es posible forzar a una app a ejecutar el código de un potencial atacante.
528. Los componentes `WebView` son ampliamente utilizados por las apps para mostrar información, actuando de clientes web (en lugar de desarrollarse una app real completa), y por tanto, permiten obtener y mostrar contenidos o páginas web proporcionadas por los servidores y aplicaciones web asociados a la app.
529. Existen dos métodos críticos asociados a los componentes `WebView`. Por un lado, el método `"setJavaScriptEnabled()"`, que indica si se permite la ejecución de código JavaScript dentro de la `WebView`⁴⁸.
530. Pese a que por defecto este método no está habilitado, es decir, los componentes `WebView` no ejecutan código JavaScript por defecto, esta funcionalidad suele ser habilitada por los desarrolladores de apps a través del método `"setJavaScriptEnabled()"`.
531. Por otro lado, el método `"addJavascriptInterface()"` proporciona elevadas capacidades al motor (y código) de JavaScript para invocar e interactuar con otros componentes y métodos (código) de Android. Por tanto, se recomienda sólo hacer uso del mismo por parte de las apps cuando se tenga plena confianza en el código HTML y JavaScript a ejecutar.
532. Las versiones de Android 4.1 (API 16) e inferiores son vulnerables frente a la posibilidad de inyección de código JavaScript que permita la ejecución de comandos de sistema operativo, ya que se permite el uso de esta funcionalidad de manera generalizada y por defecto (CVE- 2012-6636).
533. Durante los años 2011, 2012 y 2013 [Ref.- 62] se publicaron los detalles del impacto y las vulnerabilidades asociadas al método `"addJavascriptInterface()"` de `WebView` en dispositivos Android ≤ 4.1 (API ≤ 16), incluyendo la posibilidad de ejecución remota de código (RCE, *Remote Code Execution*). Un potencial código JavaScript malicioso puede

⁴⁷ <http://developer.android.com/reference/android/webkit/WebView.html>

⁴⁸ <http://developer.android.com/reference/android/webkit/WebSettings.html>

hacer uso de las capacidades de interacción entre JavaScript y Java (reflexión, Java Reflection APIs), y al ser procesado por un componente WebView, ejecutar comandos de sistema operativo y código nativo a través de una shell (mediante el método "exec()" de Java y la invocación de, por ejemplo, "/system/bin/sh").

534. Esta vulnerabilidad no sólo afectaba a los componentes WebView de las apps de terceros de Android, sino también a apps nativas como el navegador web existente por defecto, por ejemplo en Android 4.1.2, al disponer del método "addJavascriptInterface()" habilitado por defecto (CVE-2013-4710) [Ref.- 151], o incluso en Android 4.2 y 4.3 (Jelly Bean) (CVE-2014-1939)⁴⁹.
535. La criticidad de esta vulnerabilidad es muy elevada, debido al número de componentes y apps que pueden verse afectados, y a la existencia de exploits disponibles públicamente (por ejemplo, en entornos ofensivos como Metasploit⁵⁰).
536. Versiones posteriores de Android, como Android 4.2 o superiores, también pueden ser vulnerables si las apps hacen uso de WebView y del método "addJavascriptInterface()" con compatibilidad para versiones anteriores de Android (4.1 o previas), o si por ejemplo, las apps emplean librerías de servicios y redes de publicidad y anuncios (Ads) que hacen uso de estas mismas capacidades dentro de la app. El impacto de un potencial ataque depende de los permisos y capacidades de cada app.
537. Por defecto, desde la versión 4.2 de Android (API 17), y en concreto en el caso de las apps específicamente compiladas para esa versión (y/o versiones superiores) de Android (mediante la directiva "targetSdkVersion"), la vulnerabilidad ha sido mitigada. Para evitar este tipo de ataques, los componentes WebView, aunque dispongan del método "addJavascriptInterface()" habilitado, deben declarar específicamente la anotación "@JavascriptInterface"⁵¹ para cada método que desean pueda ser invocado desde código JavaScript, y además esos métodos deben ser declarados como públicos.
538. La implementación de WebView en Android estaba basada en WebKit⁵² (motor utilizado por el navegador web por defecto de Android, Android Browser) hasta la versión Android 4.4, donde se empezó a hacer uso por defecto de Chromium⁵³ (motor utilizado por el nuevo navegador web existente por defecto en Android 4.4 o posterior, Chrome).
539. Numerosas otras vulnerabilidades de seguridad que afectan a los componentes WebView han sido descubiertas y publicadas en los últimos años, afectando a versiones de Android inferiores a 4.4 (KitKat) [Ref.- 152]

⁴⁹ <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1939>

⁵⁰ http://www.rapid7.com/db/modules/exploit/android/browser/webview_addjavascriptinterface

⁵¹ <http://developer.android.com/guide/webapps/webview.html>

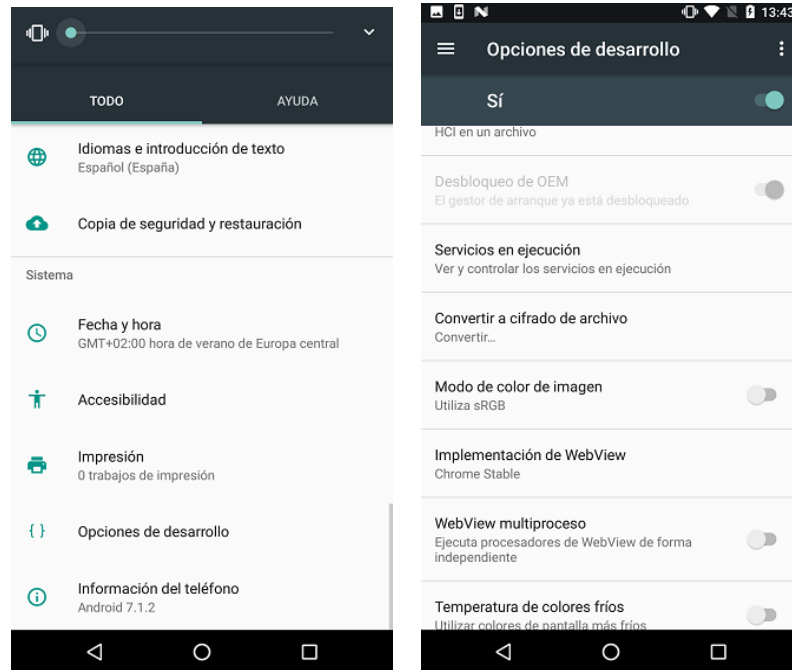
⁵² <https://www.webkit.org>

⁵³ <http://www.chromium.org>

- [Ref.- 61], tanto en su integración con JavaScript, como facilitando la utilización de técnicas de ataque para evitar la aplicación de la Same Origin Policy (SOP) en WebViews o el navegador web [Ref.- 118] [Ref.- 100], habiéndose publicado numerosas vulnerabilidades sobre la SOP (también conocidas como UXSS, Universal Cross-Site Scripting) para Android 4.3 [Ref.- 61].
540. Google confirmó en enero de 2015 que no proporcionará actualizaciones y parches de seguridad para WebView en versiones de Android previas a 4.4 [Ref.- 119].
541. Por tanto, se recomienda actualizar la versión de Android, o en caso de no ser posible, hacer uso de navegadores web que disponen de su propio motor de renderización (capacidad de procesar y mostrar contenidos web), como por ejemplo Chrome (disponible para Android 4.0 o superior) o Firefox (disponible para Android 2.3 o superior) para Android.
542. Android 5.0 incluyó una nueva versión de Chromium para las WebViews con soporte para WebRTC, WebAudio y WebGL, junto al resto de especificaciones de Web Components. A través de una nueva clase (PermissionRequest) las apps pueden dar permiso a las WebViews para acceder a otros componentes o recursos, como la cámara o el micrófono.
543. Aunque Chromium es el motor empleado para las WebViews desde Android 4.4, a partir de Android 5.0 se distribuyó como un fichero APK independiente y puede ser actualizado desde Google Play, con el objetivo de agilizar la distribución de actualizaciones que resuelvan problemas de seguridad (entre otros) que afecten al navegador web por defecto y a las apps que hacen uso de WebViews.
544. El comportamiento por defecto de las WebViews para las apps cambió en Android 5.x, bloqueándose los contenidos mezclados (es decir, cuando una misma página web hace uso de contenidos HTTPS y HTTP⁵⁴) y las cookies de terceros, si la app especifica una versión objetivo del nivel de API 21 o superior [Ref.- 503].
545. A partir de la versión 51 del navegador web existente por defecto en Android 7.x, Chrome, el APK de Chrome integra y proporciona el componente WebView, es decir, es quien se encargará de mostrar (o renderizar) los contenidos web, proporcionando su implementación para las apps que hagan uso de WebViews. Por tanto, el componente WebView se actualizará a través de Chrome. Así, en lugar de tener dos componentes o apps (Chrome y WebView), se dispondrá de uno solo, y no será necesario actualizarlos por separado. El componente WebView es un navegador web interno personalizado que permite a las aplicaciones de Android mostrar contenidos web.
546. Es posible seleccionar el proveedor de WebView a través de las opciones de desarrollo, mediante el menú " Ajustes - [Sistema] {} Opciones de desarrollo - Implementación de WebView".

⁵⁴ https://developer.mozilla.org/en-US/docs/Web/Security/Mixed_content

547. También a partir de Android 7.x y Chrome 51, el componente WebView ejecutará el contenido web dentro de un proceso contenedor (*sandbox*) independiente, siempre que la opción de desarrollador "Multiprocess WebView" se encuentre habilitada a través de "Ajustes - [Sistema] {} Opciones de desarrollo - WebView multiproceso":



548. El contexto JavaScript de las aplicaciones desarrolladas para Android 7.x se reseteará cuando se cargue una nueva página web. En las apps desarrolladas para versiones anteriores de Android, el contexto existente cuando se cargó la primera página web se transfiere a las nuevas instancias de WebView. Por tanto, los desarrolladores que quieran inyectar contenido JavaScript en las WebViews deberán ejecutar el script después de que se cargue la página web [Ref.- 702].

7.12. CONFIGURACIÓN POR DEFECTO DEL DISPOSITIVO MÓVIL

549. El dispositivo móvil está configurado por defecto con valores fijados por el fabricante del sistema operativo, el fabricante del *hardware* o el operador de telefonía móvil, que pueden desvelar detalles del terminal o de su propietario.

550. Para evitar la revelación de información sensible, se recomienda modificar esos valores existentes por defecto por valores elegidos por su propietario.

551. El proceso inicial de instalación y configuración del dispositivo móvil (ver apartado "23. Apéndice A: Proceso de instalación y configuración inicial") tiene una influencia muy limitada sobre los valores existentes por defecto y utilizados en diferentes servicios y tecnologías.

552. El nombre del dispositivo móvil Android, o *hostname* al tratarse de un dispositivo Linux, no puede ser fijado a través de los menús de los ajustes de configuración. Su valor por defecto (desde Android 2.2) es "android-<código>", donde "<código>" es un valor de 16 caracteres hexadecimales

(ver el análisis de la propiedad "net.hostname" posteriormente; por ejemplo "android-9f6edb6be8c72475").

553.El código corresponde al identificador de Android (ANDROID_ID), un valor de 64 bits (16 caracteres hexadecimales) generado aleatoriamente en el primer arranque del dispositivo móvil y que permanece constante durante toda la vida del dispositivo [Ref.- 46], es decir, hasta que se reinstale o se restablezcan los valores de fábrica.

554.Originalmente en versiones previas de Android, y desafortunadamente, el nombre del dispositivo móvil, que también es empleado como nombre para el servicio DNS (*Domain Name System*), hacía uso del carácter de subrayado ("_", *underscore*), que no es un carácter válido según el estándar del protocolo DNS y los servicios de resolución de nombres [Ref.- 92]. A partir de Android 4.0 ya se hace uso del carácter guión ("-") en su lugar.

555.Únicamente los dispositivos móviles Android en los que se ha realizado el proceso de *root* permiten la modificación del nombre del dispositivo móvil (por ejemplo, mediante el comando "hostname" de Linux, o equivalente, y la modificación de los scripts de arranque⁵⁵) [Ref.- 92], pese a tratarse de una característica solicitada a Google desde enero de 2010.

556.Las acciones de consulta también pueden ser llevadas a cabo a través del modo de depuración de Android y de la herramienta "adb", mediante la ejecución del siguiente comando, por ejemplo, desde Windows:

```
C:\> adb.exe shell getprop net.hostname
android-862d1dccb46e594c
```

557. La operación de modificación del nombre no se puede llevar a cabo con "adb", al ser necesario disponer de permisos de administración como el usuario root. "adb" no puede ejecutar como root en versiones del sistema operativo Android de producción (con un tipo de "build" igual a "user"), es decir, en dispositivos móviles reales y no emuladores, salvo en dispositivos Android *rooteados*:

```
C:\> adb.exe shell getprop ro.build.type
user
```

558. En algunas versiones no oficiales de Android, como la versión *rootada* de LineageOS⁵⁶ (previamente denominada CyanogenMod), es posible modificar de forma permanente el nombre del dispositivo móvil.

559.Las referencias internas a nombres empleadas por el dispositivo móvil Android por defecto pueden ser obtenidas a través de la utilidad "adb" y de sus capacidades de depuración para acceder a las propiedades del dispositivo móvil ("adb shell getprop"):

```
C:\> adb shell getprop | findstr name
[net.bt.name]: [Android]
[net.hostname]: [android-862d1dccb46e594c]
[persist.radio.use_cc_names]: [true]
[ro.build.version.all_codenames]: [REL]
```

⁵⁵ <https://play.google.com/store/apps/details?id=nk.bla.android.autostart>

⁵⁶ <https://lineageos.org>


```
[ro.build.version.codename]: [REL]
[ro.frp.pst]: [/dev/block/platform/soc.0/f9824900.sdhci/by-name/persistent]
[ro.product.name]: [bullhead]
```

560. En caso de desvelarse el nombre del dispositivo móvil en sus comunicaciones, como por ejemplo en el tráfico DHCP (donde se desvela el valor de la propiedad "net.hostname"; ver apartado "19.2. Seguridad en TCP/IP"), éste podría revelar detalles tanto del fabricante del dispositivo móvil, como por ejemplo el modelo del terminal y la versión concreta de Android, como del propietario. Estos datos serían muy útiles para un potencial atacante interesado en explotar vulnerabilidades en ese tipo concreto de dispositivo móvil o interesado en realizar ataques dirigidos hacia una persona u organización concreta.

561. Adicionalmente, Android puede emplear otros nombres para el dispositivo móvil al hacer uso de otros servicios y tecnologías, como por ejemplo Bluetooth (ver apartado "15. Comunicaciones Bluetooth") o el punto de acceso Wi-Fi (ver apartado "16.6. Punto de acceso o zona Wi-Fi"), por lo que la configuración de los ajustes de Bluetooth o del punto de acceso Wi-Fi permiten modificar su valor.

Nota: El valor de la propiedad denominada "net.bt.name", con el valor "Android" por defecto, no está relacionada con el nombre Bluetooth del dispositivo móvil.

562. De ser posible, en versiones futuras de Android, sería recomendable disponer de ajustes de configuración que permitan modificar el nombre del dispositivo móvil por uno que no desvele detalles ni del dispositivo móvil (fabricante o modelo) ni del propietario (evitando tanto referencias al nombre de la persona como de la organización asociada al mismo), como por ejemplo "dm0001" ("dispositivo móvil 0001").

8. GESTIÓN EMPRESARIAL DE DISPOSITIVOS MÓVILES BASADOS EN ANDROID

563. Las estrategias y soluciones empresariales de gestión de dispositivos móviles, conocidas como MDM en inglés (*Mobile Device Management*), constituyen un elemento fundamental para gestionar el nivel de riesgo y los mecanismos de seguridad en todas las organizaciones y compañías que pretendan integrar el uso de estos dispositivos en sus actividades diarias.

564. Se dispone de información detallada sobre las soluciones de gestión empresarial de dispositivos móviles o MDM en la guía "CCN-STIC-457 - Gestión de dispositivos móviles: MDM (Mobile Device Management)" [Ref.- 403].

565. En 2006, Google comenzó a proporcionar capacidades empresariales para la gestión de dispositivos móviles basados en Android a través de la plataforma Google Apps o Google Apps for Your Domain [Ref.- 13], accesibles desde un navegador web y que no requerían del despliegue de servidores dedicados para este propósito, al emplearse los servicios de Google "en la nube" asociados a un dominio propio.

566. En septiembre de 2016, Google publicó la evolución de Google Apps, posteriormente denominado Google Apps for Work, y finalmente denominado G Suite [Ref.- 205].

567. G Suite es un conjunto (o *suite*) de herramientas colaborativas (en tiempo real) y de productividad, asociadas a los servicios y productos de Google "en la nube", que no sólo proporciona capacidades empresariales de gestión de los dispositivos móviles, sino acceso a servicios como Gmail, Hangouts, Calendar, Google+ para las comunicaciones entre usuarios, Google Drive para el almacenamiento "en la nube", Google Docs, Sheets, Slides, Forms, y Sites para mejorar la productividad, etc.

568. Para poder implantar estas políticas de seguridad es necesario que el dispositivo móvil disponga de la app "Google Apps Device Policy", disponible de forma gratuita en Google Play [Ref.- 13].

569. Desde la versión 2.2 de Android los administradores de un dominio Google Apps (ediciones *Premier* o *Business*, *Government* o *Education*) o G Suite disponen de capacidades para gestionar políticas de seguridad sobre los dispositivos móviles (que van siendo ampliadas a través de las diferentes versiones de la plataforma), incluyendo [Ref.- 91]:

- Borrado remoto de los datos del dispositivo móvil (*wipe*), en caso de haber sido perdido o robado (ver apartado "10.4. Eliminación de datos del dispositivo móvil").

Nota: Este proceso es equivalente a la opción "Restablecer datos de fábrica" disponible desde el menú "Ajustes - Copia de seguridad [Datos Personales]".

- Definir el número de intentos de acceso fallidos antes de borrar los contenidos del dispositivo móvil (*wipe*).
- Definir un número de días tras el que, si el dispositivo móvil no se ha sincronizado con G Suite, será borrado (*wipe*).
- Forzar la solicitud de un código de acceso en el dispositivo móvil.
- Definir la longitud mínima del código de acceso.
- Definir la fortaleza del código de acceso mediante la utilización de letras y números.
- Definir la política del código de acceso, incluyendo los días de validez antes de que el código expire, bloquear un número concreto de códigos expirados,
- Definir el temporizador de bloqueo del dispositivo móvil tras un periodo de inactividad, cuya duración puede ser también definida en la política de seguridad.
- Auditoría de aplicaciones.
- Definir los requisitos de versión de la app "Google Apps Device Policy".
- Eliminación remota de una cuenta de usuario de un dispositivo móvil.

- Bloquear los dispositivos móviles cuya seguridad ha sido comprometida.
- 570.El administrador también puede configurar redes Wi-Fi y gestionar los certificados digitales de acceso a la red Wi-Fi a través de la app de gestión.
- 571.Adicionalmente, los usuarios de los dispositivos móviles pueden de forma remota restaurar (o resetear) el código de acceso (ver apartado "9.18. Restauración del código de acceso"), hacer que suene el dispositivo móvil, bloquear el mismo, localizar su ubicación y llevar a cabo el borrado remoto de los datos (*wipe*; si esta opción ha sido habilitada por el administrador).
- 572.Para poder localizar el dispositivo móvil es necesario disponer del permiso de localización asignado a la app de gestión, que es activado automáticamente sólo para dispositivos con un perfil de trabajo o propiedad de la organización (no en dispositivos personales) [Ref.- 91]. La información de localización no se comparte con el administrador.
- 573.La versión 2.0 (o superior) de la app "Google Apps Device Policy" permite adicionalmente a los administradores definidos en Google Apps incluir como requisito de la política de seguridad el cifrado de los datos de las *tablets* basados en Android 3.0 o los *smartphones* en la versión 4.0 (ver apartado "10.1. Cifrado del dispositivo móvil"), opción disponible desde el panel de control asociado en Google Apps [Ref.- 97].
- 574.En el caso de que el dispositivo móvil esté configurado con múltiples cuentas de G Suite, cada una haciendo uso de una política de seguridad diferente, será aplicada la política de seguridad más restrictiva. Por ejemplo, si la política de seguridad de un dominio (o cuenta) requiere códigos de acceso de 6 caracteres, y la de otro de 8 caracteres, será necesario disponer de un código de acceso de 8 caracteres [Ref.- 91].
- 575.El acceso a la gestión de dispositivos móviles y de las políticas de seguridad para los usuarios empresariales de G Suite (*Business, Government y Education*) por parte del administrador está disponible desde el panel de control de G Suite, o desde un dispositivo móvil mediante la app "Google Admin" [Ref.- 213].
- 576.Las políticas establecidas por el administrador también están disponibles para el usuario empresarial desde el propio dispositivo móvil empleando la pantalla "Policies" de la app "Google Apps Device Policy" [Ref.- 91].

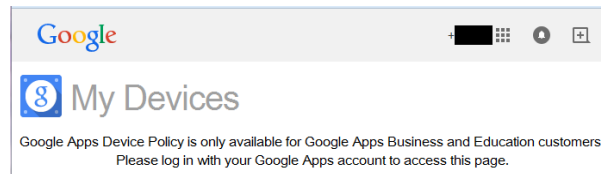
Nota: Las capacidades de gestión de dispositivos móviles Android a través de G Suite quedan fuera del alcance de la presente guía. Los administradores de G Suite disponen de documentación oficial para la gestión de los dispositivos móviles⁵⁷.

577. Una vez se establecen las políticas de seguridad a nivel empresarial y se dispone de la app instalada en los dispositivos móviles a gestionar, un usuario podría desinstalar la app. Sin embargo, el administrador puede configurar el entorno de gestión para que en ese escenario no sea posible sincronizar ningún dato entre el dispositivo móvil y el entorno corporativo (e-mails, contactos,

⁵⁷ <https://support.google.com/a/answer/1056433>

eventos de calendario o fotografías) desde el momento en que se lleve a cabo la desinstalación de la app.

578.Sin embargo, el acceso a "My Devices"⁵⁸ por parte de un usuario no asociado a un dominio de Google Apps o G Suite no está permitido, obteniéndose el siguiente mensaje informativo:



579.Microsoft, a través de Microsoft Exchange ActiveSync, también proporciona capacidades de gestión empresarial de dispositivos móviles Android [Ref.- 38], siendo posible la aplicación de numerosas políticas de seguridad [Ref.- 36].

580.Las capacidades disponibles en la política de seguridad dependen de la versión de Android existente en los dispositivos móviles, y de la versión de Microsoft Exchange empleada por la organización (2003, 2007, 2010, 2013..., y la versión de Service Pack (SP) concreta).

Nota: La gestión empresarial detallada de dispositivos móviles Android mediante Microsoft Exchange y ActiveSync (2003 SP2, 2007 o superior), Microsoft Exchange Online (<https://www.microsoft.com/online/exchange-online.aspx>) o System Center Configuration Manager (SCCM) 2012, incluyendo su implementación, configuración, e integración con las soluciones Microsoft Exchange está fuera del alcance de la presente guía [Ref.- 36][Ref.- 37].

8.1. GESTIÓN REMOTA DE DISPOSITIVOS ANDROID POR PARTE DEL USUARIO

581.El soporte para aplicaciones empresariales fue introducido por Android 2.2 a través de la API "Android Device Administration" [Ref.- 268] [Ref.- 272]. Esta API proporciona capacidades de administración a nivel de sistema, como permitir establecer políticas de contraseñas, añadir capacidades de borrado remoto y servicios de gestión del dispositivo móvil y de sus aplicaciones.

582.Las políticas de administración de dispositivos pueden estar programadas dentro de la app de gestión o solicitarse dinámicamente desde un servidor corporativo. La app de gestión se puede instalar en el dispositivo móvil del usuario final a través de Google Play, de un mercado de apps conocido y de confianza (por ejemplo, privado a la organización), o a través de un servidor web.

583.Cuando se instala la app de gestión, el sistema solicitará al usuario el permiso de "administración de dispositivos". Si se le concede, la app impondrá sus políticas, y tendrá la capacidad de tomar las acciones definidas

⁵⁸ El servicio "My Devices", accesible mediante <https://www.google.com/apps/mydevices>, ha sido reemplazado por "Encontrar mi dispositivo" (Find my devices)

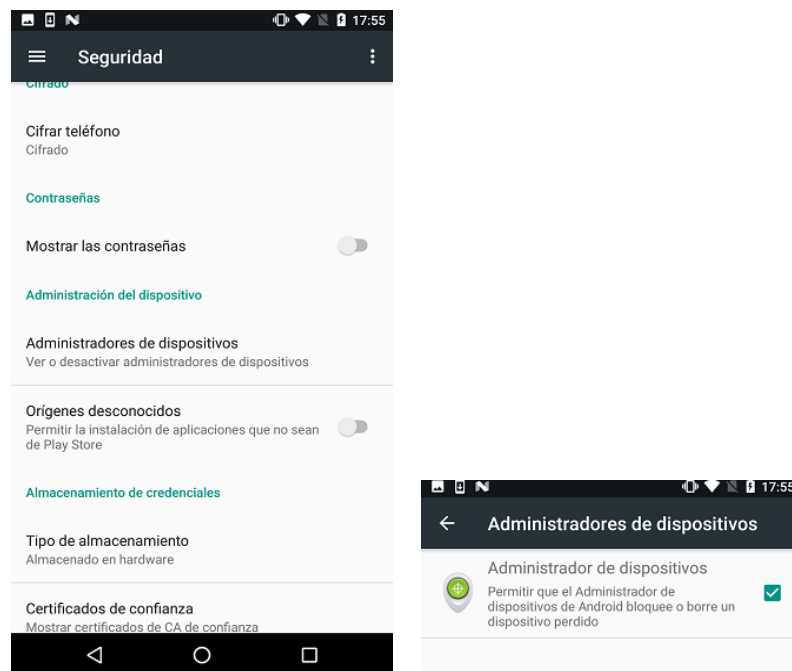
si éstas no se cumplen. Así por ejemplo, si la app define una política de longitud mínima de contraseña y el usuario no la cumple, podría denegarle el acceso a un determinado servicio.

584. Adicionalmente, la administración de dispositivos móviles puede establecer políticas que actúen ante la instalación o borrado de determinadas apps, bien sea lanzando un mensaje de aviso o bloqueando por completo la acción.

585. La API de administración de dispositivos no permite cumplimientos parciales de las políticas, es decir, sólo aquellos dispositivos que cumplan todas las restricciones se considerarán aptos.

586. Si un dispositivo contiene varias apps con el permiso de administración del dispositivo, se impondrá la que defina la política más restrictiva, ya que no existe ningún mecanismo para seleccionar una app de administración para diferentes escenarios.

587. Es posible consultar qué aplicaciones tienen el permiso de administración de dispositivos a través de "Ajustes [Personal] - Seguridad [Administración de Dispositivos] - Administradores de dispositivos":

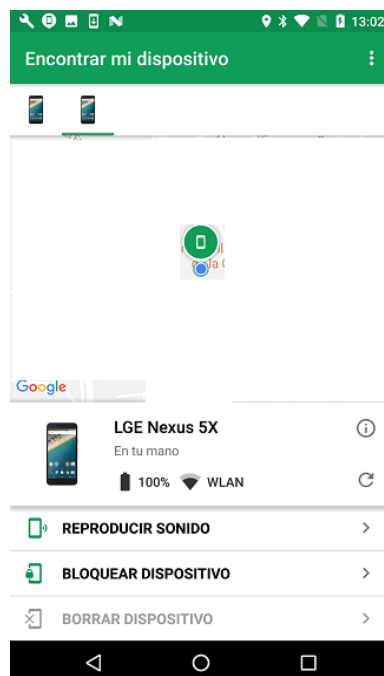


588. El permiso de administración de dispositivos, al permitir la gestión remota del mismo, incluyendo su borrado, nunca debe ser concedido salvo en escenarios de plena confianza, ya que una app maliciosa podría solicitarlo y hacer uso de estas capacidades avanzadas.

589. El acceso a la gestión remota del dispositivo móvil por parte de los usuarios particulares, no asociados a un dominio de Google Apps o G Suite, se puede llevar a cabo a través del recurso "Encontrar mi dispositivo", disponible en la URL "<https://www.google.com/android/find>", o desde la app disponible en Google Play⁵⁹, también denominada "Encontrar mi dispositivo".

⁵⁹ <https://play.google.com/store/apps/details?id=com.google.android.apps.adm>

590. Este recurso y app han sustituido a los utilizados en versiones previas de Android, ambos denominados "*Android Device Manager*"⁶⁰.
591. El servicio "Encontrar mi dispositivo" (Find my device), que hasta Android 4.4 se ofrecía como una app disponible en Google Play (antiguamente denominado Play Store), pasó a formar parte del servicio "Google Play Protect" (ver apartado "Verificación de seguridad de las apps: Google Play Protect" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]), de modo que desde Android 6.x en adelante se ofrece como un componente con permisos de "Administración de dispositivos" [Ref.- 226], y, entre otras cosas, permite al usuario desde un terminal Android encontrar otros dispositivos móviles Android, tanto *smartphones* como *tablets*. Adicionalmente, está disponible como app en la Google Play Store.
592. Es importante diferenciar entre el servicio "Encontrar mi dispositivo", y su cliente o módulo asociado presente por defecto en versiones actuales de Android (reflejado con el mismo nombre "Encontrar mi dispositivo" en la sección "Administradores de dispositivos"), y la app "Encontrar mi dispositivo", que se puede instalar desde la Play Store pero que no está instalada por defecto. El servicio interactúa con el cliente o módulo, que permite o deniega la gestión remota sobre el propio dispositivo móvil. La app actúa como interfaz gráfico, permitiendo el uso del servicio y de las capacidades de gestión remota sobre dispositivos externos que compartan con él la cuenta de usuario de Google.



593. La app "Encontrar mi dispositivo" también sigue estando disponible para Android 7.x, para el caso en que se desee localizar otro dispositivo desde el

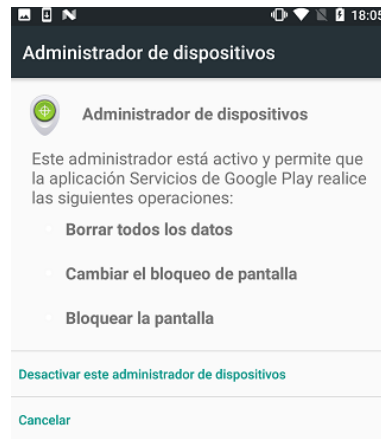
⁶⁰ La URL del recurso "Android Device Manager" era: "<https://www.android.com/devicemanager>". En el momento de elaboración de la presente guía, este enlace redirige al nuevo recurso "Encontrar mi dispositivo": "<https://www.google.com/android/find>".

terminal móvil (ver la imagen anterior). Sus opciones son similares a las ofrecidas por el servicio de gestión remota "Encontrar mi dispositivo" (disponible en <https://www.google.com/android/find>) de Google.

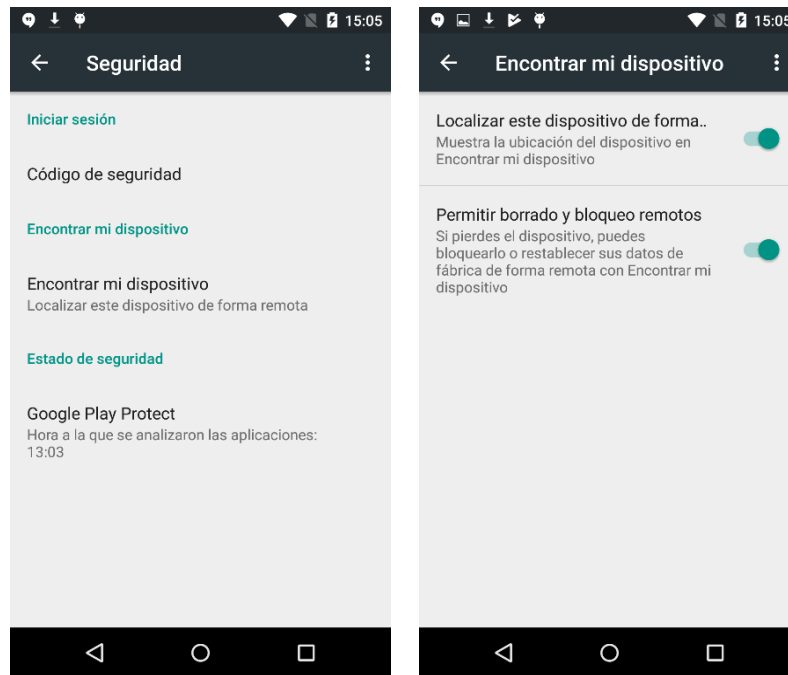
594. Es muy importante reseñar que, tanto la funcionalidad del portal de gestión remota como los mensajes que se muestran en respuesta a las diferentes acciones solicitadas a través de él, son dependientes de Google (y de sus servicios asociados) y pueden variar a lo largo del tiempo. Las opciones y mensajes descritos en la presente guía corresponden a noviembre de 2017.

595. Los requisitos para hacer uso de la funcionalidad de gestión remota son [Ref.- 60]:

- Disponer de una conexión de datos: para poder hacer uso de las capacidades de gestión remota es necesario que el dispositivo móvil disponga de una conexión a una red Wi-Fi o de datos móviles (2/3/4G), a fin de poder comunicarse con los servidores de Google y responder a las acciones solicitadas por estos.
 - Si el dispositivo móvil no dispone de una conexión de datos, el administrador de dispositivos Android no podrá bloquear la pantalla, modificar el código de acceso (referido por Google como la contraseña de bloqueo de la pantalla), hacer sonar el dispositivo móvil, o borrar sus datos remotamente hasta que se establezca una conexión a través de una red Wi-Fi o de la red de datos móviles. Al seleccionar una acción en el administrador de dispositivos Android, ésta se llevará a cabo la próxima vez que el dispositivo móvil vuelva a estar conectado. Lógicamente, el administrador de dispositivos Android no funciona para dispositivos móviles que estén apagados o con el modo avión activado.
- Disponer del administrador de dispositivos Android activo: desde el menú "Ajustes [Personal] - Seguridad [Administración de Dispositivos] - Administradores de dispositivos [Personal] Encontrar mi dispositivo" es posible activar esta funcionalidad, y la pantalla de confirmación informará de las capacidades de gestión asociadas. Las capacidades de gestión remota que se obtienen al activar el administrador de dispositivos Android son: *bloquear la pantalla* (equivale a bloquear remotamente un dispositivo móvil perdido o robado), *cambiar el bloqueo de pantalla* (en caso de que no se hubiese configurado un código de acceso en el dispositivo móvil, desde el interfaz web de gestión remota se puede configurar uno; si ya hubiese uno vigente, la configuración de un nuevo código de acceso no modificará el código actual) y *borrar todos los datos* (equivale a una restauración a fábrica remota, o *wipe*).

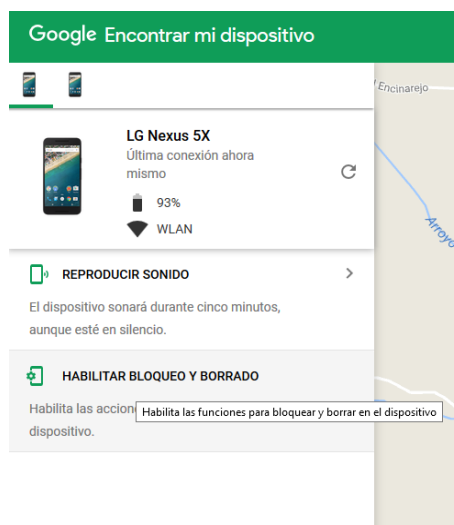


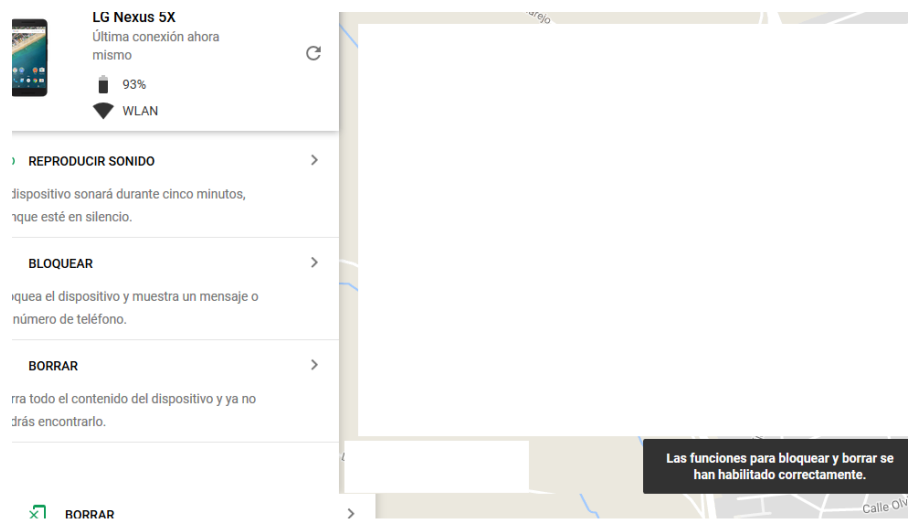
- Disponer de una sesión iniciada con la cuenta de Google del usuario desde el dispositivo móvil, disponible automáticamente en caso de haber sido configurada dicha cuenta.
- El dispositivo móvil debe estar visible en Google Play: el dispositivo se puede hacer visible desde la sección "Visibilidad" de Google Play ("<https://play.google.com/settings>"). Si un dispositivo móvil está oculto en Google Play, no aparecerá en la funcionalidad de "Encontrar mi dispositivo".
- Tener el servicio de ubicación (o localización) activo: desde la app "Ajustes de Google", mediante el menú "[Servicios] Ubicación", es posible habilitar la opción "Ubicación" (servicios de localización) mediante el botón "Sí".
 - Este acceso es equivalente al acceso estándar correspondiente a los servicios de localización, disponible desde "Ajustes - [Personal] Ubicación".
- En la versión de Android 5.x, la app "Google" existente por defecto permitía modificar las capacidades y configuración del administrador de dispositivos Android a través de la opción "[Servicios] Seguridad - [Administrador de dispositivos] Administrador de dispositivos". Las opciones de configuración posibles, que se podían habilitar y deshabilitar individualmente, y que se muestran en las imágenes inferiores, eran "Localizar este dispositivo de forma remota" y "Permitir borrado y bloqueo remotos":



596.Sin embargo, esta capacidad de permitir sólo la opción deseada no es configurable en la versión de Android 7.x, y las dos opciones son aceptadas implícitamente si se habilita el administrador de dispositivos "Encontrar mi dispositivo".

597.Sin embargo, aunque la opción "Ajustes [Servicios] - Seguridad - [Administrador de dispositivos] Administradores de dispositivos - Encontrar mi dispositivo" esté deshabilitada en el dispositivo móvil, el interfaz web de gestión remota permite habilitar estas capacidades de forma remota. Si se selecciona esta opción remotamente desde el interfaz web, se activará de nuevo el interruptor "Encontrar mi dispositivo" en el dispositivo móvil. En el interfaz web se mostrará el mensaje *"Las funciones para bloquear y borrar se han habilitado correctamente"*. Es decir, estas capacidades son gestionadas remotamente desde el interfaz web y no desde el propio dispositivo móvil:





598. La información sobre los dispositivos asociados a una cuenta de Google está disponible en el menú "Controles de la actividad de tu cuenta - Información de los dispositivos", y, pese a que podría pensarse que esta información se limitaría a datos ligados solo al terminal, en realidad incluye datos que no son propios del dispositivo móvil, como los contactos o el calendario de la cuenta de Google vinculada a él [Ref.- 235].

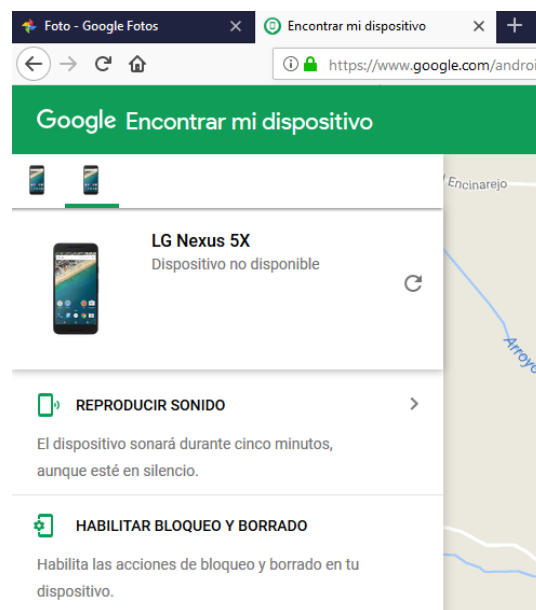
599. Si se desea acceder a las capacidades de gestión remota ante la pérdida o robo del dispositivo móvil, se dispone de dos opciones con funcionalidad similar, ambas descritas en detalle a continuación:

- Utilizar el recurso "Encontrar mi dispositivo" (opción 1), disponible en "<https://www.google.com/android/find>" (o "<https://android.com/find/>").
- Acceder a la cuenta de Google del usuario del dispositivo móvil desde "<https://myaccount.google.com>" y seleccionar la opción "Encontrar tu móvil" (opción 2), que mostrará la lista de dispositivos móviles que están (o han estado) vinculados a esa cuenta, pudiéndose seleccionar el dispositivo deseado.

600. Si se hace uso del recurso "Encontrar mi dispositivo" (opción 1) [Ref.- 201] para acceder a las capacidades de gestión remota, tras introducir las credenciales de la cuenta de Google del usuario, se mostrarán las siguientes opciones (ver la imagen inmediatamente anterior):

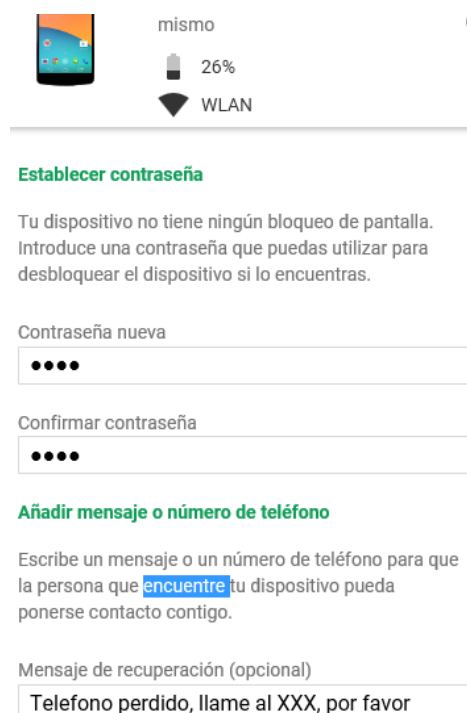
- Reproducir sonido: esta opción hace que el dispositivo móvil emita un sonido con el máximo volumen para poder localizar su ubicación, incluso si se encontraba en silencio. El sonido se reproducirá durante 5 minutos o hasta que se interaccione físicamente con el dispositivo, a través de los botones físicos o de la pantalla.
- Bloquear: esta opción bloquea la pantalla del dispositivo móvil (en caso de estar desbloqueada), y adicionalmente permite que se muestre un mensaje y/o número de teléfono para contactar con su propietario.
- Borrar: esta opción permite eliminar remotamente todos los datos del dispositivo móvil, siendo equivalente a una restauración a fábrica (*wipe*).

601. Si la opción de localización remota está activa, al seleccionar la flecha circular disponible a la derecha de los datos del dispositivo móvil, se refrescará la información de ubicación del mismo. La ubicación, mostrada en Google Maps, reflejará en el mapa dónde se encuentra el dispositivo actualmente ("Última conexión ahora mismo").
602. A fecha de elaboración de la presente guía, en caso de que el dispositivo no esté localizable cuando se solicita su ubicación, se mostrará un mensaje indicando que no está disponible. En el pasado, al intentar localizar un dispositivo que no estaba disponible, se mostraba la fecha y hora de la última vez que se recibió información de su ubicación y dónde se encontraba el dispositivo móvil en ese momento.

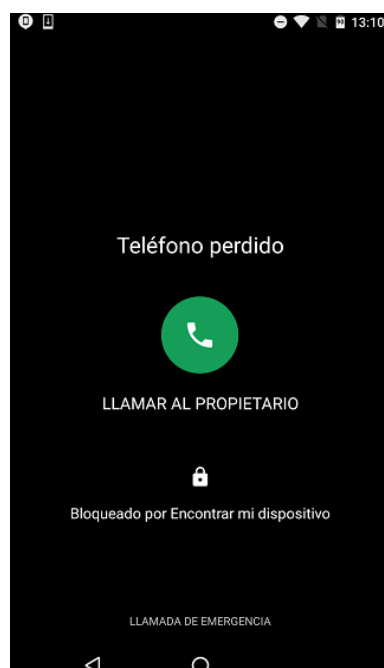


603. La opción "Bloquear" mostrará un menú que permite introducir un código de acceso (o contraseña) y proporcionar un mensaje que se mostrará en la pantalla de bloqueo del dispositivo móvil. La primera vez que se pulse esta opción, se muestra un mensaje indicando que el dispositivo no tiene ningún bloqueo de pantalla. Este mensaje puede resultar confuso, pues no implica que el dispositivo móvil realmente no tenga configurado un código de acceso actualmente, sino que no se ha añadido aún ninguno a través del interfaz web de gestión remota.
604. Por otro lado, aunque se introduzca un código de acceso distinto al que ya estuviese vigente en el dispositivo móvil, el código de acceso actual no se reemplazará y seguirá siendo válido para desbloquear el terminal. Por tanto, esta opción únicamente tiene validez cuando el dispositivo móvil no dispone de un código de acceso configurado, opción no recomendada desde el punto de vista de seguridad.
605. Al confirmar la recepción de la operación, se mostrará el mensaje "*Bloqueo solicitado*". Cuando el dispositivo móvil vuelva a disponer de conectividad de datos, recibirá la petición, y ejecutará la acción.
606. El interfaz web mostrará un mensaje que indicará si la operación tuvo éxito (se mostrará el mensaje "*Bloqueado*") o no; sin embargo, en el momento de

elaboración de la presente guía, en múltiples ocasiones se ha recibido el mensaje "*No se ha podido bloquear el dispositivo*" aunque la operación haya concluido con éxito (por lo que no se debe confiar en la fiabilidad actual a la hora de confirmar el éxito de las operaciones).



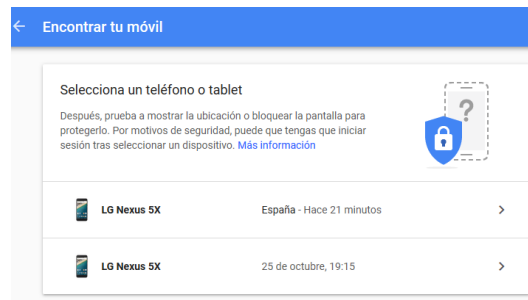
607. Como resultado, si la acción de bloqueo ha tenido éxito, el dispositivo móvil mostrará una nueva pantalla de desbloqueo, con el mensaje especificado previamente (cada vez que se apague la pantalla). Para proceder a desbloquearlo, hay que pulsar en el botón de retroceder a la pantalla previa (o flecha hacia la izquierda situada en la parte inferior de la pantalla) que permitirá acceder a la pantalla principal de desbloqueo.



608. Si el dispositivo móvil no tenía un código de acceso configurado y se solicitó habilitarlo remotamente, será necesario introducir el nuevo código de acceso, y, si ya disponía de él, habrá que introducir el código ya existente.
609. La opción "Borrar" eliminará todos los datos de usuario del dispositivo móvil y restaurará a fábrica el terminal, dejando la versión del sistema operativo Android disponible en el momento de llevar a cabo esta acción. El proceso de eliminación de los datos es equivalente a un *hard reset*, es decir, al restablecimiento del dispositivo móvil a la configuración de fábrica. Durante el proceso, por tanto, se eliminarán las apps instaladas, junto a todos los datos y configuraciones realizadas por el usuario.

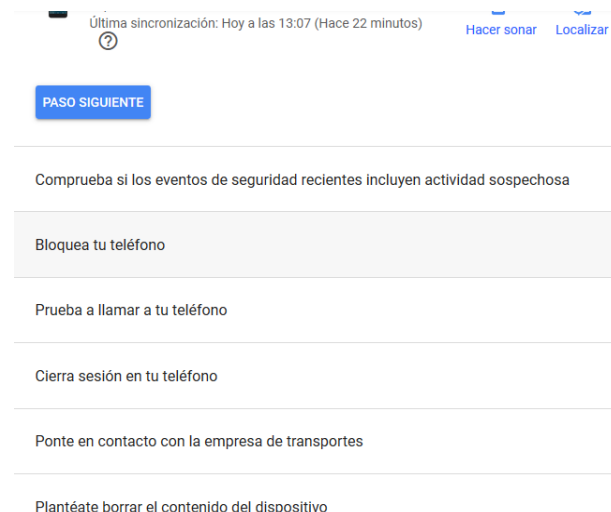


610. Las peticiones de bloqueo y borrado remotas se encolan en los servidores de Google, por lo que, si el dispositivo móvil se encuentra apagado o sin conexión de datos (Wi-Fi o móviles) cuando se envía la petición, la misma será reenviada cuando el dispositivo disponga de una conexión de datos.
611. Es importante tener en cuenta que puede existir una demora entre que se ordena una determinada acción desde el interfaz web de gestión remota y su ejecución efectiva en el dispositivo móvil. Aunque normalmente se debería recibir un mensaje que confirme el éxito o fracaso de la operación, desafortunadamente se ha constatado que, en determinadas ocasiones, los mensajes de confirmación de la operación en la interfaz web de gestión remota no siempre son coherentes con la acción realizada, no se muestran o tienen un contenido erróneo. Sin embargo, la tasa de coherencia en Android 6.x es significativamente superior a la observada en Android 5.x.
612. Si se utiliza la cuenta de Google del usuario vinculada al dispositivo móvil, mediante la opción de gestión remota "Encontrar tu móvil" (opción 2), y tras introducir las credenciales de la cuenta de Google del usuario, se mostrará un menú con las opciones recomendadas para localizarlo una vez seleccionado el dispositivo móvil:



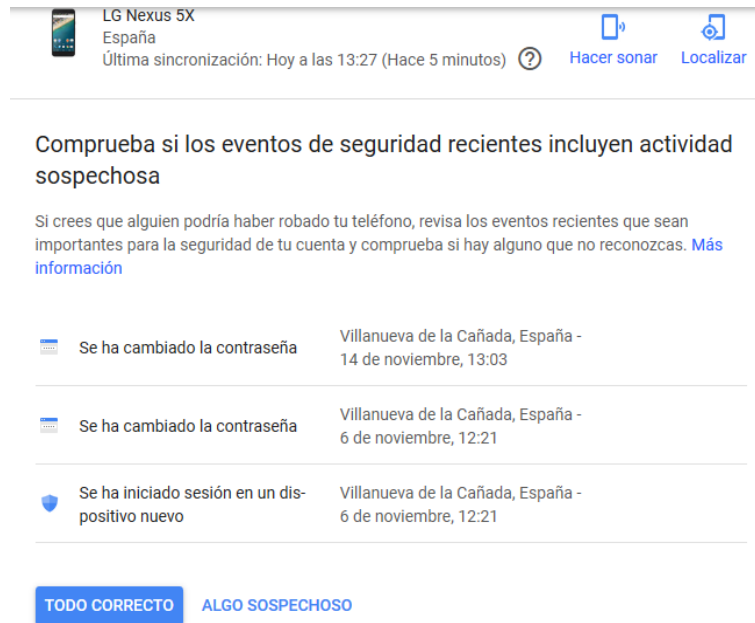
613. Se recomienda por defecto seguir las múltiples opciones disponibles en el orden en el que son presentadas, pues se parte de la premisa de que el dispositivo móvil podría estar aún próximo al usuario.

614. Sin embargo, la política de seguridad para dispositivos móviles de la organización debería definir una secuencia de actuación diferente para este tipo de escenarios en función de los requisitos de seguridad de los dispositivos móviles y de la criticidad de los datos que gestionan.



615. Si desde este interfaz web de gestión se selecciona la opción "Localizar", se abrirá el enlace "<https://www.google.com/android/find>" (equivalente al recurso "Encontrar mi dispositivo" correspondiente a la opción 1 descrita previamente).

616. Si se selecciona la opción "Paso siguiente", se desplegará un menú que muestra los últimos eventos de seguridad registrados en la cuenta de Google del usuario, como por ejemplo un cambio de contraseña o un inicio de sesión en un nuevo dispositivo, desde el que no se había efectuado un inicio de sesión nunca.



LG Nexus 5X
España
Última sincronización: Hoy a las 13:27 (Hace 5 minutos) ? [Hacer sonar](#) [Localizar](#)

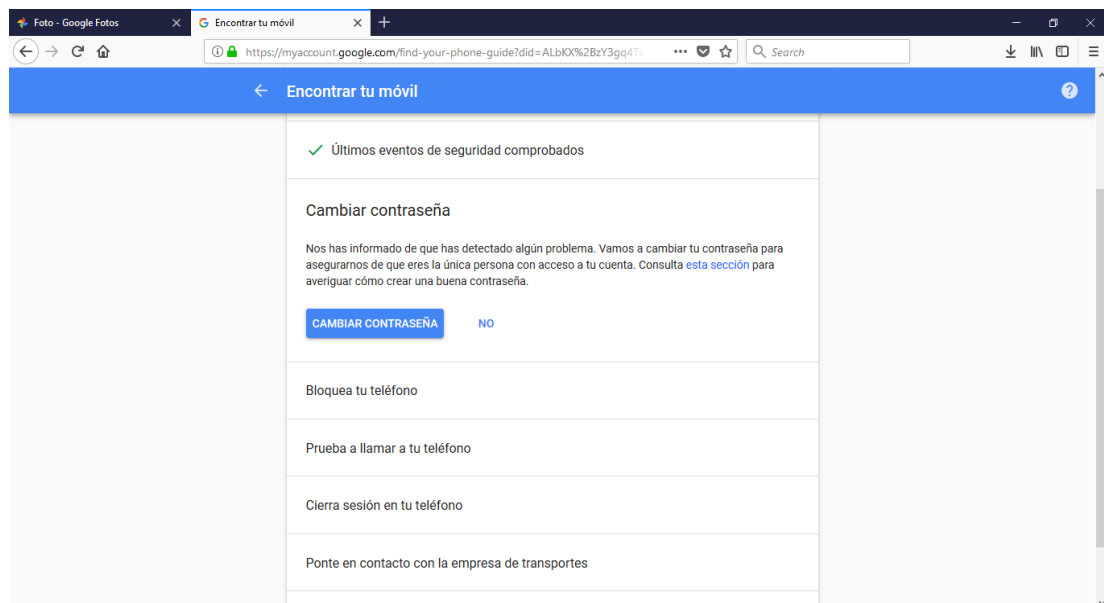
Comprueba si los eventos de seguridad recientes incluyen actividad sospechosa

Si crees que alguien podría haber robado tu teléfono, revisa los eventos recientes que sean importantes para la seguridad de tu cuenta y comprueba si hay alguno que no reconozcas. [Más información](#)

Se ha cambiado la contraseña	Villanueva de la Cañada, España - 14 de noviembre, 13:03
Se ha cambiado la contraseña	Villanueva de la Cañada, España - 6 de noviembre, 12:21
Se ha iniciado sesión en un dispositivo nuevo	Villanueva de la Cañada, España - 6 de noviembre, 12:21

TODO CORRECTO [ALGO SOSPECHOSO](#)

617. Si el usuario indica que hay algo sospechoso, se propondrá un cambio de contraseña de la cuenta de Google para evitar comprometerla:



Encontrar tu móvil

✓ Últimos eventos de seguridad comprobados

Cambiar contraseña

Nos has informado de que has detectado algún problema. Vamos a cambiar tu contraseña para asegurarnos de que eres la única persona con acceso a tu cuenta. Consulta [esta sección](#) para averiguar cómo crear una buena contraseña.

CAMBIAR CONTRASEÑA NO

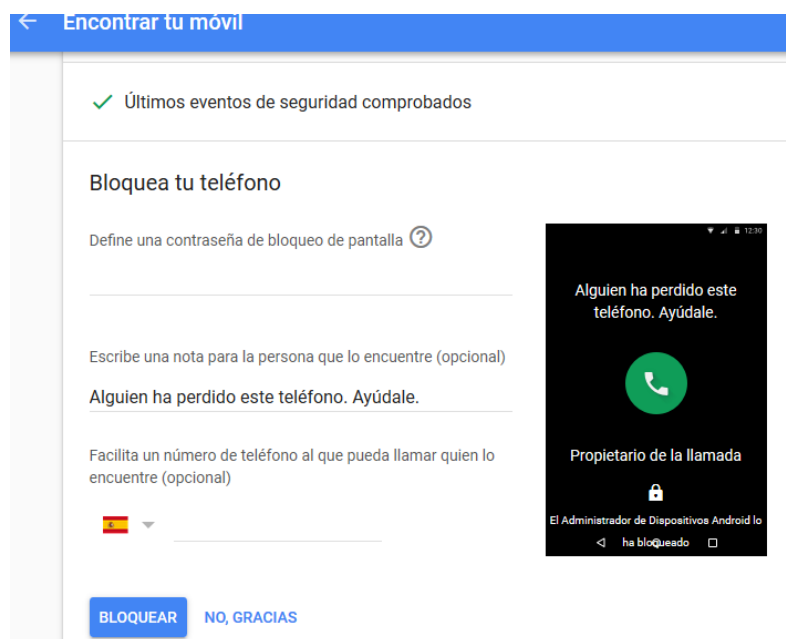
Bloquea tu teléfono

Prueba a llamar a tu teléfono

Cierra sesión en tu teléfono

Ponte en contacto con la empresa de transportes

618. Si, por el contrario, no se identifica ninguna actividad extraña, se desplegará la opción "Bloquea tu teléfono" (equivalente a la opción "*Bloquea tu teléfono*" del listado de múltiples opciones disponibles), que permite bloquear remotamente el dispositivo móvil. Además, si el dispositivo móvil no disponía de un código de acceso previamente establecido (escenario desaconsejado desde el punto de vista de seguridad), se configurará un código de acceso para el bloqueo del dispositivo con el valor especificado en este menú; si el dispositivo ya disponía de un código de acceso establecido, tal como se ha descrito previamente para la opción 1, se mantendrá el código actualmente configurado en el dispositivo, tal como muestra el mensaje que se obtiene mediante la ayuda (icono con el símbolo "?"):



619.El hecho de que no se lleve a cabo la modificación del código de acceso (o contraseña) actual en el dispositivo móvil supone una diferencia notable respecto al comportamiento del servicio y app *Android Device Manager* empleado en versiones previas de Android, en la que el código de acceso sí se cambiaba por el nuevo valor, tal como muestra la siguiente imagen:



620.Una vez ejecutada la acción, si ésta ha tenido éxito, se recibe confirmación de la operación en el interfaz web de gestión remota. Si no ha sido posible contactar con el dispositivo móvil, se informa de que se ha enviado la solicitud mediante el mensaje "*Hemos enviado una solicitud de bloqueo a tu dispositivo correctamente*". Cuando el dispositivo móvil vuelva a disponer de conectividad de datos, recibirá la petición, y ejecutará la acción. Como se ha descrito previamente, esto implica que los servidores de Google encolan las peticiones destinadas al dispositivo, y las envían cuando detectan que el

dispositivo móvil está de nuevo disponible. En ese momento se mostrará el mensaje de confirmación⁶¹:

El teléfono está bloqueado. Como ya tienes un bloqueo de pantalla configurado, no se utilizará la contraseña que acabas de configurar.

621. Como resultado, si la acción de bloqueo ha tenido éxito, el dispositivo móvil mostrará una nueva pantalla de desbloqueo, con el mensaje especificado previamente (cada vez que se apague la pantalla). En dicha pantalla se mostrará una nota que indica "Bloqueado por el 'Encontrar mi dispositivo'".
622. La opción "*Prueba a llamar a tu teléfono*" no realiza acciones directas sobre el dispositivo móvil, sino que propone al usuario alternativas para localizar el dispositivo, diferentes a las que se pueden utilizar desde la página web de gestión. Igualmente ocurre con la opción "*Ponte en contacto con la empresa de transportes*" (referida a "ponerse en contacto con el operador de telefonía móvil").
623. La opción "*Cierra sesión en tu teléfono*" provoca que se cierre la sesión asociada a la cuenta de Google del usuario en el dispositivo móvil. De ese modo se impedirá que, en caso de poderse desbloquear el dispositivo por parte de un tercero no autorizado, éste tenga acceso a los datos de la cuenta. En el dispositivo móvil se mostrará la notificación "Acción necesaria en cuenta" y, al desbloquearse el dispositivo y abrir la notificación, se indicará "Has cerrado sesión en tu cuenta de Google". Al seleccionar el botón "Siguiente", se solicitará al usuario la contraseña de la cuenta de Google que estaba activa antes de cerrarse la sesión (aunque haya varias configuradas en el dispositivo móvil).

Cierra sesión en tu teléfono

Cierra la sesión de tu cuenta de Google @gmail.com en tu teléfono perdido para que nadie pueda acceder a ella. Podrás hacer sonar, localizar y bloquear tu teléfono, así como borrar su contenido.

CERRAR SESIÓN

NO, GRACIAS

624. Por último, la funcionalidad "*Plantéate borrar el contenido del dispositivo*" ofrece al usuario la posibilidad de eliminar todos los datos del terminal remotamente desde el interfaz web de gestión. Una vez ejecutada la acción, mediante el botón "Sí, Borrar", si el dispositivo móvil disponía de conectividad de datos, se restaurará su configuración de fábrica:

⁶¹ Desafortunadamente, no se ha identificado la existencia de ningún registro (o *log*) que refleje las diferentes acciones llevadas a cabo mediante los mecanismos de gestión remota, junto a las confirmaciones del dispositivo móvil, incluyendo una referencia temporal.

Plantéate borrar el contenido del dispositivo



Al borrar tu dispositivo, se impedirá que otras personas vean tu contenido. Si decides hacerlo, ten en cuenta lo siguiente:

- Es posible que no podamos borrar por completo todas las tarjetas de memoria de tu dispositivo.
- La función Encontrar tu móvil y el Administrador de Dispositivos Android ya no podrán localizar, hacer sonar ni bloquear tu teléfono.
- Perderás de forma permanente el acceso a toda la información del dispositivo de la que no se haya hecho ninguna copia de seguridad en Google.

El contenido del teléfono se borrará la próxima vez que se conecte a Internet. Si no lo hace, no se borrará.

SÍ, BORRAR

NO, GRACIAS

¿Quieres borrar todos los datos?

Mediante esta acción, se restablece la configuración original de fábrica en tu dispositivo. Tus aplicaciones, fotos, música y configuración se eliminarán. Tras borrar el dispositivo, el Administrador de Dispositivos Android dejará de funcionar. Esta acción no se puede deshacer. Es posible que no podamos borrar el contenido de la tarjeta SD del dispositivo.

Si tu dispositivo no está conectado a Internet, restableceremos la configuración original de fábrica cuando se conecte.

CANCELAR

SÍ, BORRAR

625. Tras confirmar la operación de borrado, se intentará contactar con el dispositivo:

Intentando conectar con tu dispositivo para borrar su contenido...

626. Tan pronto se tenga de nuevo conexión de datos desde el dispositivo móvil, se enviará la orden de borrado y se informará al usuario con el mensaje "Se ha borrado el dispositivo". En la pantalla del terminal, se mostrará un mensaje "Restableciendo datos de fábrica" y el terminal se apagará y reiniciará mostrando el menú de configuración inicial.

✓ Prueba a llamar a tu teléfono

❓ Cierra sesión en tu teléfono

✓ Ponte en contacto con la empresa de transportes

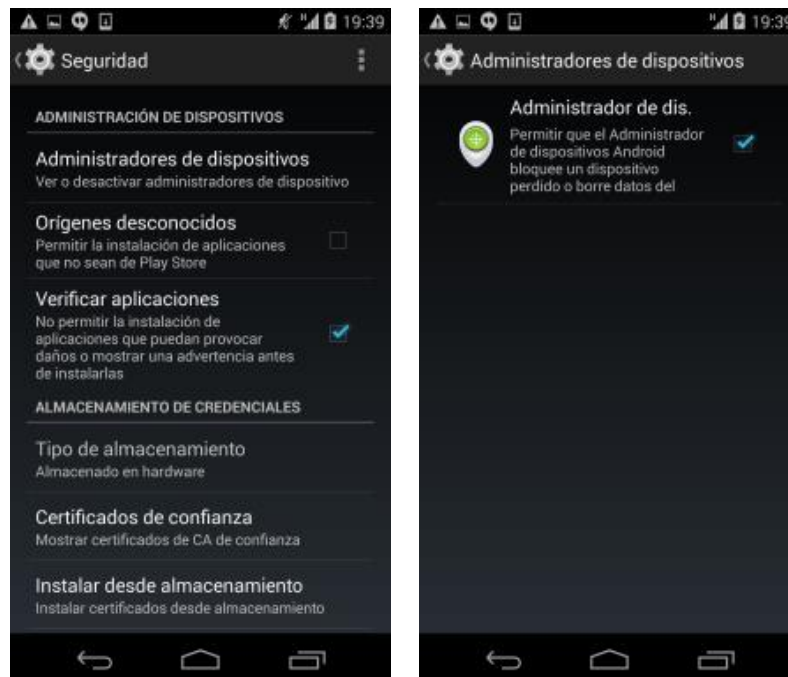
✓ Se ha borrado el dispositivo

627. En el pasado, tras lanzarse la acción de borrado del dispositivo móvil, el administrador de dispositivos enviaba un mail a la cuenta de Google del usuario del dispositivo informando de que se la solicitud de borrado de datos había sido enviada y que el administrador de dispositivos Android intentaría borrar los datos del dispositivo y de la tarjeta SD (el contenido de

ese mensaje se ilustra a continuación). A fecha de elaboración de la presente guía, dicho mensaje no es enviado:

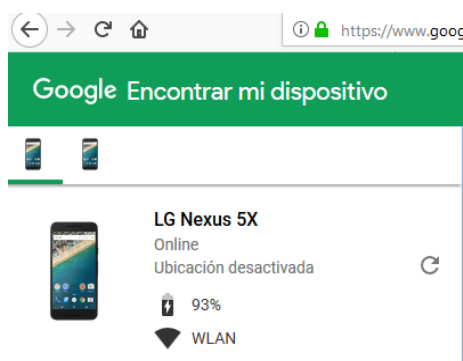


628. Otros dispositivos móviles con versiones de Android más antiguas, anteriores a la versión Android 5.x, también pueden disponer del administrador de dispositivos Android (aunque pudiera no estar habilitado por defecto), ya que el mismo fue introducido por Google en una de las actualizaciones del componente (y app) asociado a los Servicios de Google Play⁶² (sin ser necesario realizar una actualización de la versión del sistema operativo):

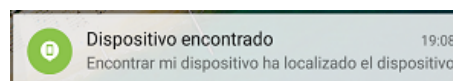


⁶² <http://www.howtogeek.com/179638/not-getting-android-os-updates-heres-how-google-is-updating-your-device-anyway/>

629. Desde Android 5.x, el administrador de dispositivos Android está habilitado por defecto.
630. En el caso de dispositivos móviles con varios usuarios, únicamente el usuario administrador podrá habilitar o deshabilitar el administrador de dispositivos "Encontrar mi dispositivo".
631. Si el servicio de ubicación del dispositivo móvil estuviese desactivado en "Ajustes - [Personal] Ubicación", la opción "Localizar (este dispositivo de forma remota)" se mostraría inhabilitada y el interfaz web de gestión remota informaría de este hecho mediante el mensaje "Ubicación desactivada":



632. En versiones anteriores de Android, como la 5.1.1, al realizarse un cambio en la configuración de las opciones de ubicación remota desde el propio dispositivo móvil, el interfaz web de gestión podía no reflejar el cambio inmediatamente y seguir mostrando el mensaje de "Ubicación desactivada". En este caso, será necesario salir de la interfaz web de gestión y volver a entrar para refrescar la información del estado del dispositivo móvil.
633. Las acciones realizadas desde el interfaz web generan notificaciones en el centro de notificaciones del dispositivo móvil. Por ejemplo, cuando se accede a la opción de localización y se solicita encontrar el dispositivo, en el terminal se mostrará una notificación "Dispositivo encontrado - Encontrar mi dispositivo ha encontrado ...". Si se ha solicitado el bloqueo del dispositivo, también se mostrará una notificación en el terminal con el contenido del mensaje de bloqueo. Estas notificaciones podrían desvelar a un usuario no autorizado las tareas de gestión realizadas.



634. Recientemente, Google incorporó a su interfaz de administración de la cuenta de usuario de Google (concretamente, en el apartado "Inicio de sesión y seguridad"), el apartado "Actividad de los dispositivos y eventos de seguridad", en el cual se pueden ver y gestionar eventos que, en caso de haberse extraviado el dispositivo móvil, permiten detectar si se está realizando algún acceso desde él en la cuenta de Google del usuario. Para obtener más detalles sobre este servicio, se recomienda consultar el capítulo "Inicio de sesión y seguridad" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407].

8.2. ANDROID FOR WORK

635. La filosofía BYOD (*Bring Your Own Device*), consistente en permitir que los trabajadores de una organización puedan emplear sus dispositivos móviles personales para acceder a los recursos corporativos y llevar a cabo tareas propias de su puesto de trabajo, requiere la puesta en marcha de políticas de seguridad que protejan tanto los recursos físicos y lógicos, como los datos.
636. Se incorpora en Android el concepto de EMM (Enterprise Mobility Management), como solución de gestión empresarial de dispositivos móviles. Google, además de su propio servicio G Suite, reconoce una serie de empresas cuyas soluciones han sido certificadas para gestionar las políticas empresariales sobre dispositivos móviles Android⁶³. Asimismo, Google recomienda una serie de dispositivos móviles para uso corporativo⁶⁴.
637. Android for Work [Ref.- 197] es un entorno BYOD centrado en que los usuarios puedan aumentar su productividad laboral o profesional, permitiéndoles utilizar en el entorno de trabajo sus dispositivos móviles Android, incluyendo las mismas herramientas que usan en su vida móvil personal. Android for Work permite separar las apps de trabajo de las apps personales, con el objetivo de independizar ambos mundos, el personal y el de trabajo, no sólo a nivel de los datos, sino también de la funcionalidad.
638. El término Android for Work pasó a denominarse simplemente Android en diciembre de 2016 [Ref.- 514], al considerar Google que la funcionalidad de Android for Work ya estaba disponible en la mayoría de dispositivos móviles Android, pudiéndose emplear todos ellos en entornos de trabajo de manera estándar. Igualmente Play for Work pasó a denominarse simplemente Google Play.
639. Para poder definir, configurar y aplicar un perfil de trabajo en Android 5.x, o versiones posteriores de Android, es necesario disponer de una solución empresarial de gestión de dispositivos móviles (EMM), junto a una app asociada instalada en el dispositivo móvil (que actuará como controlador de las políticas de los dispositivos móviles empresariales). El perfil de trabajo permite definir múltiples opciones en la política de seguridad empresarial, similares a las descritas previamente para las soluciones MDM (ver apartado "8. Gestión empresarial de dispositivos móviles basados en Android").
640. Las APIs y la funcionalidad de Android for Work, disponibles en Android y en Google Play para gestionar de manera consistente y uniforme los dispositivos móviles Android, fueron integradas en el sistema operativo Android y en las APIs estándar de esta plataforma móvil a partir de esa fecha.
641. Desde Android 5.x Google ha extendido significativamente el soporte empresarial de Android, protegiendo los datos empresariales y los contenidos privados haciendo uso de zonas de almacenamiento separado para los datos de trabajo y personales.

⁶³ <https://androidenterprisepartners.withgoogle.com/#!/results/browse-all/0>

⁶⁴ <https://androidenterprisepartners.withgoogle.com/#!/results/browse-all/2>

642. Android 5.x integró en el entorno de trabajo (o *workplace*) un nuevo entorno que permite la gestión de tareas de aprovisionamiento [Ref.- 505], iniciadas por una app de administración del dispositivo (*Device Administrator*). Este administrador de dispositivo puede crear un perfil gestionado o de trabajo (*managed* o *work profile*), que se añade de forma complementaria al perfil personal del usuario del dispositivo (si éste creó uno previamente), habilitando el uso de BYOD en entornos corporativos.
643. Las notificaciones tanto del perfil personal como de trabajo, así como las tareas o apps recientes en ejecución, se muestran en una vista unificada, aunque los datos gestionados por cada perfil se mantienen aislados del otro perfil, incluso aunque una misma app esté disponible en ambos perfiles.
644. En el caso de los dispositivos móviles propiedad de la organización pero que pueden también ser utilizados en el plano personal (COPE, *Corporately-Owned Personally-Enabled*), el administrador de la organización puede instalar una app propietaria del dispositivo (*Device Owner*) en los nuevos terminales, y distribuirlos preinstalados con ésta [Ref.- 505].
645. Esta app es una versión especializada de una app de administración del dispositivo (*Device Administrator*) que tiene capacidades especiales para añadir o eliminar usuarios adicionales, y para gestionar los ajustes globales de configuración. Solo puede existir una instancia de este tipo de app por dispositivo móvil.
646. La app (o, en su caso, una app administradora del dispositivo) se encargará de instalar un perfil gestionado o de trabajo (*managed* o *work profile*) en el dispositivo móvil, que coexistirá con la cuenta personal del usuario.
647. Un perfil de trabajo permite a la organización definir y administrar los datos y apps corporativas a nivel empresarial, coexistiendo con el resto de datos personales del usuario en el mismo dispositivo móvil [Ref.- 516].
648. Mediante los perfiles de trabajo, los administradores de la organización pueden gestionar de forma más segura el entorno de trabajo, independientemente del entorno personal, y sin restringir la utilización de otros datos y apps personales por parte de los usuarios.
649. Es posible verificar si un dispositivo móvil dispone de un perfil de trabajo instalado desde el menú "Ajustes - Cuentas". En el caso de disponer de un perfil administrado, éste se mostrará en la sección de trabajo [Ref.- 516].
650. Para garantizar la seguridad del dispositivo, desde Android 5.0 se permitió a los administradores proteger los datos asociados al perfil de trabajo imponiendo un método de acceso al dispositivo que cumpla ciertas restricciones [Ref.- 520].
651. A partir de Android 7.x, se incluye la posibilidad de establecer un control de acceso específico para ejecutar las apps y acceder a los datos ligados al perfil de trabajo [Ref.- 732]. Este control de acceso (denominado *work profile security challenge*) se invocará cuando el usuario intente abrir alguna app asociada al perfil de trabajo. También se permite al usuario establecer un bloqueo unificado.

652. En caso de disponer de apps que son gestionadas dentro del perfil de trabajo por el administrador de la organización, denominadas apps gestionadas, los iconos asociados se muestran en el Launcher con un indicador de trabajo⁶⁵ (icono naranja con un maletín blanco)⁶⁶:



653. El administrador del dispositivo puede restringir el acceso a apps del sistema desde el perfil gestionado o de trabajo, como por ejemplo el acceso a la cámara.

654. Cada perfil dispone de su propio espacio de almacenamiento, por lo que no se recomienda que las apps compartan ficheros a través de mensajes (o *intents*), ya que la referencia a un fichero que es válida en una app no tiene por qué serlo en otra (si está asociada a otro perfil).

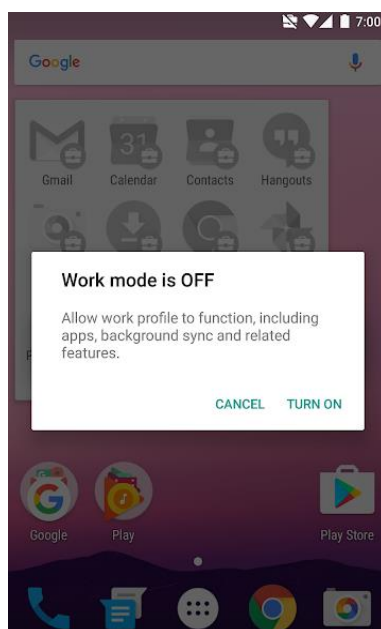
655. En su lugar, se deben emplear referencias a contenidos (no a ficheros). Así, por ejemplo, el administrador de dispositivo puede establecer que la captura de imágenes sea gestionada por la app de Cámara en el perfil personal, y la referencia a contenido especificará dónde se puede almacenar la foto. Ésta corresponderá a una ubicación donde podrá escribir la app Cámara y desde donde podrá leer la app que inició la captura de la imagen (mediante un *intent*) [Ref.- 503].

656. Adicionalmente, Android 6.x añadió soporte para múltiples capacidades de Android for Work (algunos de ellos detallados a continuación), como controles para dispositivos mono-usuario propiedad de una organización, instalación y desinstalación silenciosa de apps y certificados digitales, aceptación automática de actualizaciones de sistema operativo, o notificaciones del estado del entorno de trabajo, para que el usuario pueda identificar fácilmente que está trabajando dentro de una app gestionada (o de trabajo) o que hay tareas de apps pertenecientes al perfil gestionado (o de trabajo) ejecutando de fondo.

⁶⁵ <https://developer.android.com/about/versions/lollipop.html#Work>

⁶⁶ <https://www.wired.com/2015/02/googles-android-work-gives-phone-split-personality/>

657. Android 6.x incorporó nuevos cambios en Android for Work, como la posibilidad de ver las llamadas asociadas a contactos de trabajo en el histórico de llamadas de la app de "Teléfono", incluso desde el contexto personal, o todo un conjunto de cambios en el comportamiento asociado a las APIs disponibles para los gestores de la política de seguridad del dispositivo móvil, por ejemplo, por parte de una app administradora del dispositivo [Ref.- 602].
658. Asimismo, Android 6.x incorporó controles avanzados para la app propietaria del dispositivo en dispositivos móviles empleados en entornos COSU (*Corporate-Owned, Single-Use*), pudiendo deshabilitar (o habilitar de nuevo) el keyguard, la barra de estado, el arranque seguro (safe boot), etc [Ref.- 603].
659. Android 7.x (Nougat) introduce el denominado "modo trabajo" (*work mode*), que posibilita deshabilitar las apps asociadas al perfil de trabajo y ocultar sus notificaciones cuando se desee.



660. Android 7.x incorpora funcionalidades que afectan al comportamiento y diseño de las apps dirigidas a Android for Work, entre las que se incluyen modificaciones en la instalación de certificados, reseteo de claves, gestión de usuarios secundarios y acceso a los identificadores del dispositivo [Ref.- 730]. Las apps se ven afectadas en la siguiente medida:
- Antes de configurar el DPC (*Device Policy Controller*) se requiere instalar un certificado delegado [Ref.- 729].
 - Los administradores de dispositivo sólo pueden fijar una clave si el dispositivo no tiene una ya habilitada.
 - Cuando el dispositivo se encuentra en modo "device owner" (propietario del dispositivo), se habilita la restricción "DISALLOW_ADD_USER", de modo que los usuarios no puedan crear usuarios secundarios.

- Las apps con rol de propietario de dispositivo pueden acceder a los identificadores de dispositivo, por ejemplo, la dirección MAC asociada al interfaz Wi-Fi.
 - En Android 7.x, el acceso mediante huella se gestiona por usuario. Si un cliente DPC se basa en la API 23 o inferior, las apps de trabajo no podrán acceder a la huella. Si el DPC se basa en la API 24 o superiores, el usuario puede asociar su huella para el perfil de trabajo a través de "Ajustes - Seguridad - Seguridad del perfil de trabajo".
 - Si se instala un certificado cliente con clave privada, el certificado de la CA que lo haya emitido no se guardará en el almacenamiento de credenciales de confianza.
 - Los administradores del perfil de trabajo pueden suspender el acceso a apps que no cumplen con las políticas establecidas.
 - Se proporciona la posibilidad de configurar los dispositivos a través de aprovisionamiento mediante un código QR.
661. Por otro lado, se permite la instalación y desinstalación silenciosa de apps por parte de la app propietaria del dispositivo, sin la intervención del usuario, junto a la selección silenciosa por parte de la app propietaria del dispositivo de certificados digitales empresariales cuando una app solicita que se seleccione un certificado, sin que el usuario sea preguntado al respecto. Esta app también puede delegar en otra app la gestión de certificados digitales.
662. Las capacidades empresariales permiten obtener remotamente datos de uso de Android. La app propietaria del perfil puede consultar los datos del perfil que gestiona, mientras que la app propietaria del dispositivo puede acceder a los datos de uso del usuario principal gestionado.
663. Adicionalmente, la política de seguridad definida por la app propietaria del dispositivo puede establecer que se instalen automáticamente las actualizaciones de sistema operativo, o posponer su aceptación (incluso por parte del usuario), hasta un periodo de 30 días. Es posible también definir la ventana de tiempo diaria en la que se deberían aplicar las actualizaciones, por ejemplo, cuando el dispositivo móvil (si es utilizado en un quiosco) no está siendo utilizado.
664. Tanto la app propietaria del perfil como la app propietaria del dispositivo pueden establecer una política de permisos para gestionar todos los permisos en tiempo de ejecución de Android 6.x solicitados por las apps instaladas en el dispositivo móvil. El perfil se encargará de permitir y/o denegar las solicitudes de permisos silenciosamente, o de preguntar al usuario. Si el perfil se encarga de la gestión de ciertos permisos, el usuario no podrá posteriormente modificar la elección realizada por el perfil para dichos permisos desde "Ajustes".
665. Se ofrece la posibilidad de forzar a que las apps de trabajo siempre se conecten a través de una VPN específica, activando dicha VPN en el arranque del dispositivo (esta opción se conoce como "Always on VPN"). También los usuarios pueden configurar clientes VPN que soporten el modo

"Always on VPN" a través de "Ajustes - [Conexiones inalámbricas y redes] ... [Más] - VPN".

666. Las notificaciones asociadas dependen de la configuración de la red VPN. Para la app propietaria del perfil, las notificaciones dependen de si el perfil ha sido configurado como gestionado, personal o ambos, y para la app propietaria del dispositivo las notificaciones dependen de si la VPN se ha configurado para todo el dispositivo móvil.

667. Por otro lado, los perfiles de configuración Wi-Fi que han sido añadidos por una app propietaria de un perfil son eliminados si se borra dicho perfil del dispositivo móvil.

668. Asimismo, se dispone de la posibilidad de bloquear las configuraciones Wi-Fi añadidas por una app propietaria del dispositivo, de manera que el usuario no podrá borrarlas ni modificarlas. El usuario, sin embargo, sí podrá añadir o modificar las configuraciones de otras redes Wi-Fi. Por otro lado, la app propietaria del dispositivo tiene el privilegio para poder borrar o modificar cualquier configuración Wi-Fi, incluso las que no han sido creadas por ella.

669. Android 7.x permite definir mensajes en la configuración que simplifican la comprensión de las políticas aplicadas.

9. ACCESO FÍSICO AL DISPOSITIVO MÓVIL - INTERFAZ DE USUARIO

670. La posibilidad de disponer de acceso físico al dispositivo móvil, incluso sólo temporalmente, permitiría a un potencial atacante acceder a sus contenidos, o hacer uso de los servicios de telefonía móvil, o cualquier otra funcionalidad o servicio disponibles.

671. Por tanto, proteger el acceso al dispositivo móvil es un elemento crítico para garantizar la seguridad y la privacidad del usuario.

672. Los controles de acceso principales de cara a proteger el dispositivo móvil son:

- Establecer un PIN en la tarjeta SIM, el cual impedirá que se pueda hacer uso de las capacidades de telefonía del terminal si no se ha introducido correctamente.
- Establecer un código de acceso al propio dispositivo, que impida acceder a las opciones de configuración, a los datos y a las aplicaciones instaladas.

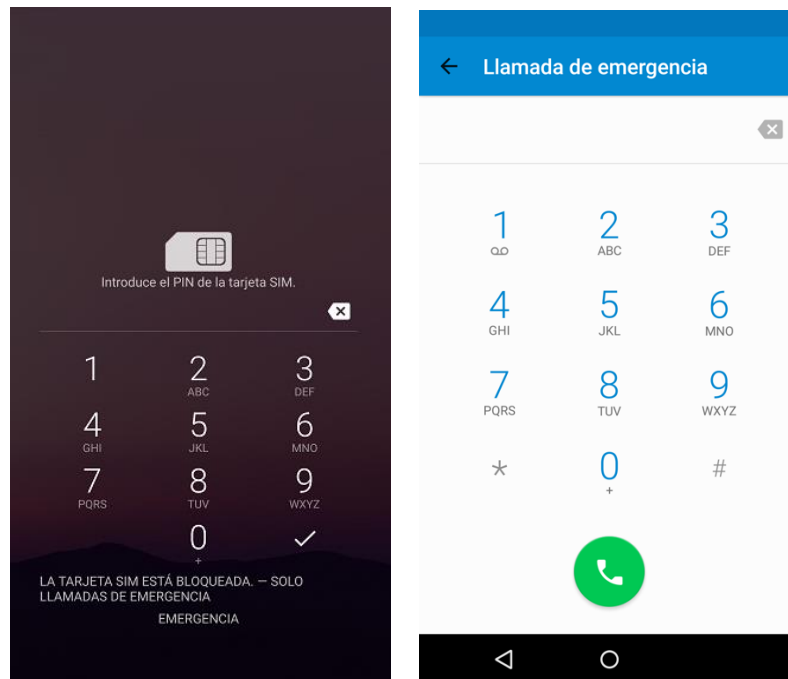
673. Para evitar ambos tipos de acceso es posible fijar un método de acceso tanto en la tarjeta SIM (mediante un PIN, *Personal Identification Number*) como en el propio dispositivo móvil.

9.1. PIN DE LA TARJETA SIM

674. El PIN de la tarjeta SIM (*Subscriber Identity Module*) bloquea el acceso no autorizado a los servicios y capacidades de telefonía móvil asociados a la propia tarjeta SIM. Si la tarjeta SIM está bloqueada (se desconoce el código

PIN), únicamente se permite la realización de llamadas de emergencia (mediante el número de teléfono 112).

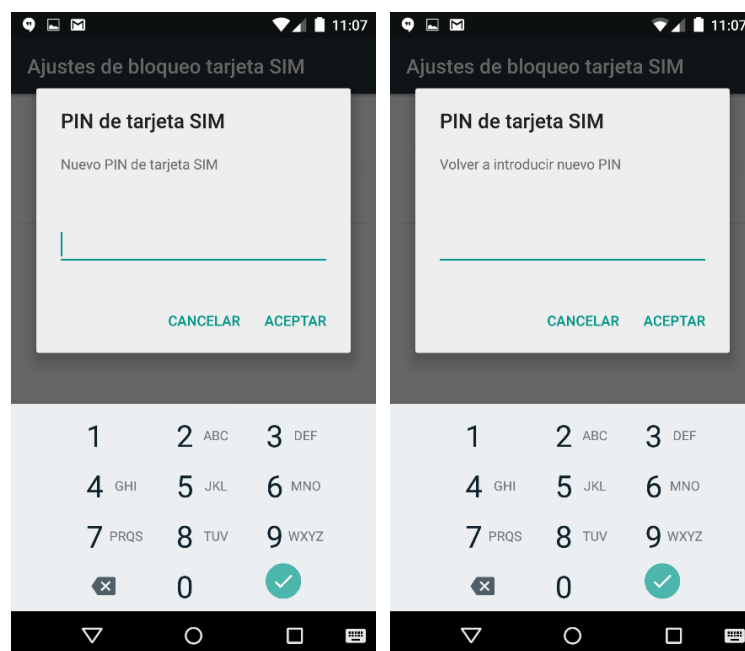
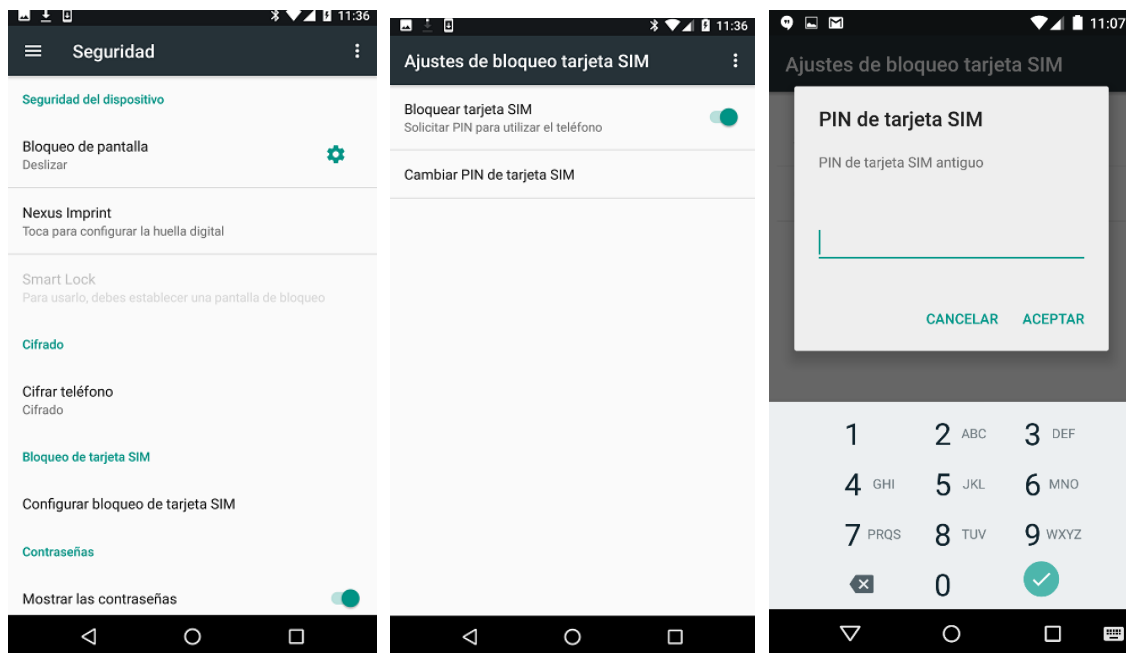
675. Sin embargo, si sólo se establece el PIN en la tarjeta SIM y no en el dispositivo móvil (ver apartado "9.2. Código de acceso al dispositivo móvil"), aún es posible para un potencial atacante acceder al terminal directamente, incluyendo sus datos y apps, tras extraer físicamente la tarjeta SIM del dispositivo móvil:



676. Si se ha establecido un PIN en la tarjeta SIM, al arrancar el dispositivo móvil no es posible acceder al mismo mediante ningún botón, ya que Android continúa solicitando el PIN de la tarjeta SIM para permitir el acceso al terminal, siendo necesario extraer la tarjeta SIM previamente para disponer de acceso al terminal sin introducir el PIN de ésta.
677. La disponibilidad de un PIN por defecto asociado a la tarjeta SIM depende del operador de telefonía móvil, siendo muy habitual que las tarjetas SIM tengan fijado un valor por defecto que es proporcionado en la documentación del contrato del operador de telefonía móvil facilitada al usuario al solicitar sus servicios.
678. El estándar de la industria empleado en la actualidad define un código de 4 dígitos (0-9) como PIN o código de acceso para proteger la tarjeta SIM, sin embargo, se recomienda hacer uso de un PIN de mayor longitud, en concreto, de 8 dígitos (longitud máxima permitida).
679. Android permite emplear un PIN de la tarjeta SIM con un valor de entre 4 y 8 dígitos.
680. Pese a la debilidad de un PIN de 4 dígitos frente a ataques de adivinación o fuerza bruta, los operadores de telefonía móvil y fabricantes de dispositivos móviles implementan otros mecanismos para evitar este tipo de ataques, como el bloqueo de la tarjeta SIM tras tres intentos de acceso fallidos. En

ese caso, para poder desbloquear la tarjeta SIM es necesario emplear una segunda clave de acceso denominada PUK (*PIN Unlock Key*).

681. Los dispositivos móviles Android 7.x, y posteriores, permiten forzar la utilización de un PIN para la tarjeta SIM (opción "Configurar bloqueo de tarjeta SIM"), así como fijar o modificar el PIN asociado a la tarjeta SIM (opción "Cambiar PIN de tarjeta SIM"), a través del menú "Ajustes [Personal] - Seguridad [Bloqueo de tarjeta SIM] - Configurar bloqueo de tarjeta SIM":



682. Se recomienda por tanto fijar en la tarjeta SIM la restricción que requiere introducir un PIN para hacer uso de ésta, y modificar el valor por defecto del PIN, fijando el PIN de la tarjeta SIM con un valor o secuencia de números que no sea fácilmente adivinable, es decir, excluyendo valores típicos como 0000, 1111 ó 1234.

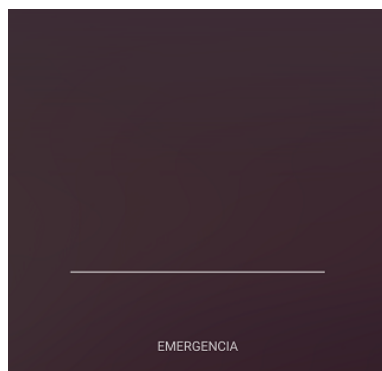
683. Debe tenerse en cuenta que cada vez que el dispositivo móvil sale del "Modo avión" (es decir, cuando este modo es desactivado; menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Modo avión"), no se solicitará el PIN de la tarjeta SIM al usuario (tal y como ocurría en versiones previas de Android), al activarse de nuevo los servicios de telefonía móvil. El valor del PIN ha sido almacenado en memoria (o cacheado) para poder disponer de nuevo de acceso a la tarjeta SIM sin que sea necesario introducir el PIN.

9.2. CÓDIGO DE ACCESO AL DISPOSITIVO MÓVIL

684. El código de acceso (PIN, contraseña, patrón de acceso, etc.) del dispositivo móvil bloquea el acceso no autorizado al terminal, incluyendo sus datos, capacidades de comunicación y aplicaciones móviles (o apps).

685. El patrón (o gesto) fue empleado como método original de desbloqueo en Android, y la posibilidad de usar un PIN o contraseña se añadió en Android 2.2.

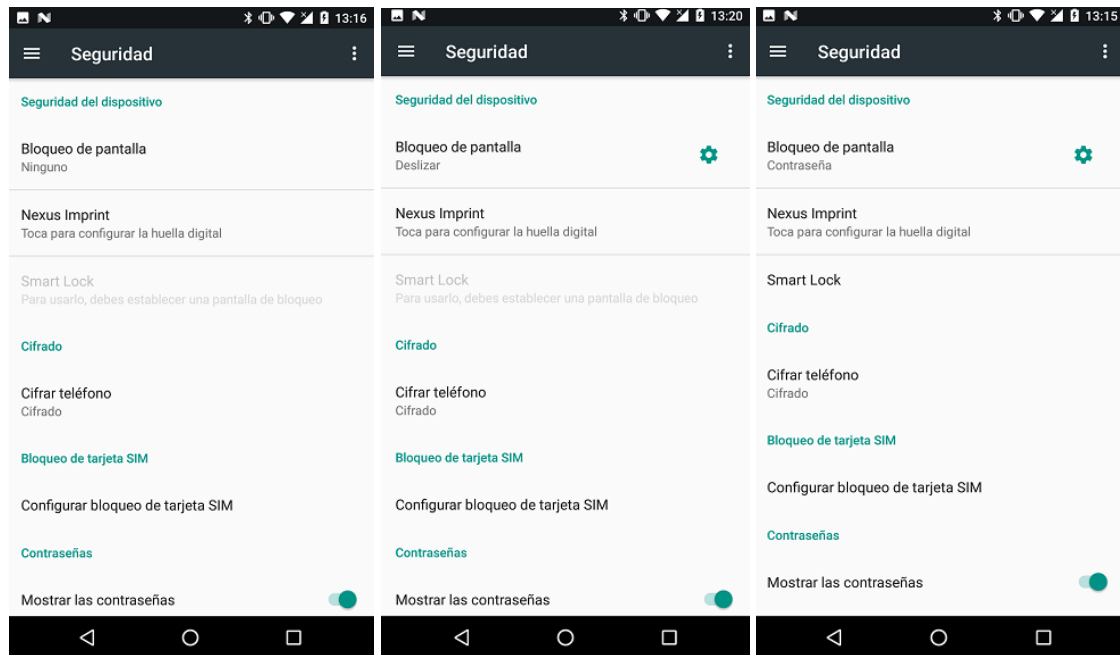
686. Una vez el dispositivo móvil está bloqueado, únicamente se permite la realización de llamadas de emergencia (112) sin el código de acceso del terminal. Para ello es necesario marcar el número de teléfono de emergencias tras pulsar el texto "Emergencia" (situado en la parte inferior de la pantalla).



687. Sin embargo, si únicamente se establece el código de acceso en el dispositivo móvil y no un PIN en la tarjeta SIM (ver apartado "9.1. PIN de la tarjeta SIM"), aún sería posible para un potencial atacante acceder físicamente al terminal, extraer la tarjeta SIM y hacer uso de la misma en otro terminal, incluidos sus servicios y capacidades de telefonía móvil asociados.

688. Por defecto, los dispositivos móviles basados en Android no obligan a fijar un código de acceso para evitar accesos no autorizados al terminal, si bien determinadas funciones sólo estarán disponibles si se ha definido un código de acceso (por ejemplo, la gestión del almacenamiento de credenciales descrita en el apartado "11. Gestión de certificados digitales y credenciales").

689. Se recomienda por tanto establecer un código de acceso al dispositivo móvil a través del menú "Ajustes [Personal] - Seguridad [Seguridad del dispositivo] - Bloqueo de pantalla":



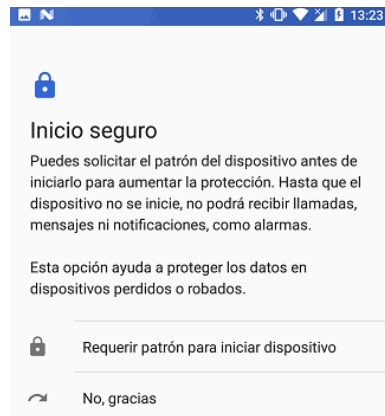
690. Android permite establecer múltiples tipos de códigos de acceso en la pantalla de desbloqueo, también conocida como *keyguard* en Android. El método que se encuentre activo en un momento concreto es el que aparece bajo "Ajustes [Personal] - Seguridad - Seguridad de la pantalla - Bloqueo de pantalla" y, para cambiarlo, el dispositivo solicitará confirmación del código de acceso actual (de estar uno vigente).

691. "Sin seguridad" (o "Ninguno"): deshabilita por completo la pantalla de desbloqueo, pudiéndose acceder directamente a toda la funcionalidad del dispositivo móvil.

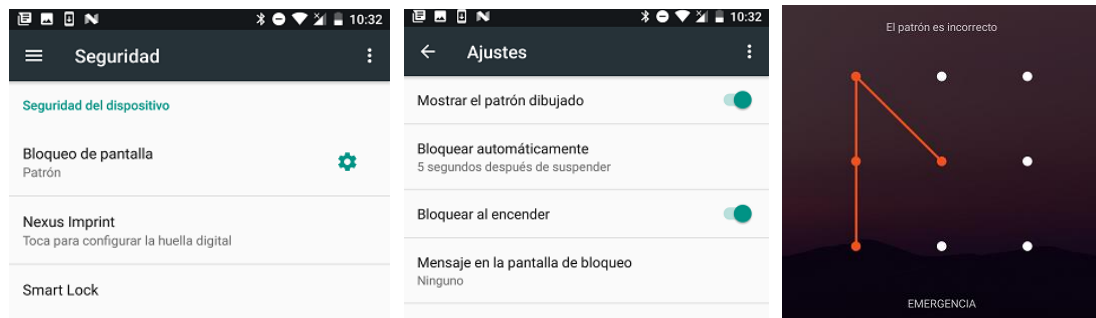
692. "Deslizar": no establece ningún código de acceso, siendo necesario únicamente deslizar el icono con un candado abierto situado en la parte inferior de la pantalla de desbloqueo hacia arriba (hacia la parte superior de la pantalla). Este icono de candado abierto informa además de que el teléfono no tiene establecida pantalla de bloqueo, lo cual podría favorecer que alguien lo sustrajese.



693. "Patrón": permite establecer un patrón (o gesto) de desbloqueo de pantalla, mediante la definición de un patrón o secuencia de movimientos (en una matriz de 3x3 puntos ubicada en la pantalla del dispositivo) que permitirá desbloquear el dispositivo móvil. El patrón debe de comprender un mínimo de 4 puntos y un máximo de 9, aplicándose una serie de reglas (como, por ejemplo, no siendo posible pasar por encima de un punto y que éste no forme parte del patrón).

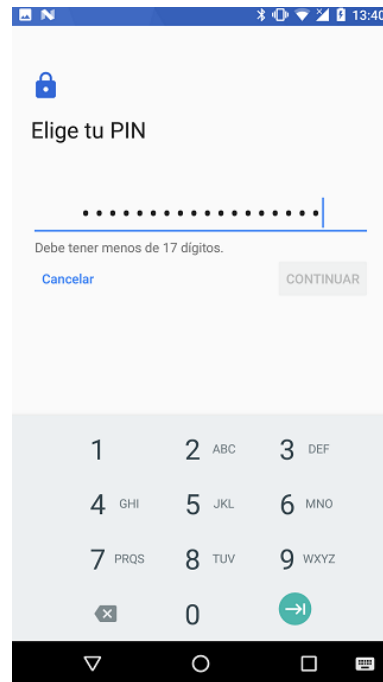
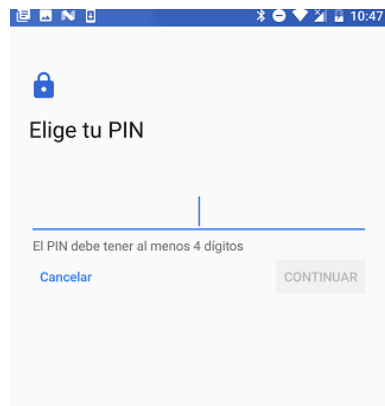


694. Se recomienda seleccionar la opción "Requerir patrón para iniciar dispositivo", a fin de proteger el dispositivo móvil antes de que se realice el arranque del sistema operativo.
695. Al seleccionar la opción "Patrón", Android proporciona la ayuda necesaria para que el usuario aprenda a crear un patrón de desbloqueo, siendo necesario que el patrón incluya al menos cuatro puntos de los nueve (matriz de 3x3 puntos) en los que se divide la pantalla del dispositivo móvil.
696. El número máximo de puntos posibles en el patrón de desbloqueo es de nueve, ya que no es posible repetir la selección de un mismo punto de la matriz de 3x3.
697. Con las anteriores restricciones, el número máximo de combinaciones posibles para los patrones de desbloqueo de Android es de 389.112 [Ref.-208].
698. Se recomienda emplear patrones que no correspondan con iniciales del nombre de usuario o letras comunes del abecedario, como por ejemplo C, L, M, N, O, P, etc., práctica que resulta común.
699. En caso de seleccionar un patrón de desbloqueo, es posible definir si el patrón es o no visible en la pantalla durante el proceso de desbloqueo del dispositivo móvil mediante la selección del icono de ajustes situado a la derecha de "Bloqueo de pantalla - Patrón" y la opción "Mostrar el patrón dibujado" (habilitada por defecto).



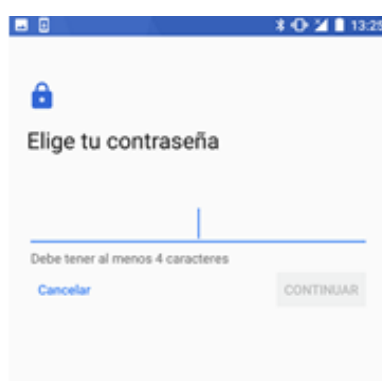
700. Aunque se deshabilite la opción de mostrar el patrón dibujado, en Android 7.x, al igual versiones anteriores, el dispositivo emitirá una respuesta táctil durante la introducción del patrón, consistente en una pequeña vibración al seleccionar cada punto de la matriz de 3x3 puntos, lo que permite al usuario confirmar que está introduciendo el patrón correcto.
701. Desde el punto de vista de seguridad se recomienda no habilitar el uso de un patrón visible (que dibujaría las líneas de unión entre los puntos que conforman el patrón), para evitar que alguien con acceso visual temporal sobre la pantalla del dispositivo móvil durante el proceso de desbloqueo pueda obtener el patrón empleado.
702. En principio, para evitar ataques de fuerza bruta, se recomendaría emplear patrones de desbloqueo complejos, que hagan uso de un mayor número de puntos, un mínimo de 6 a 9 puntos (y en ningún caso del número mínimo de tan sólo 4 puntos). No obstante, como se explica a continuación, existen ataques que, sin emplear fuerza bruta, permitirían averiguar con un alto índice de probabilidad patrones más complejos.
703. Un estudio llevado a cabo conjuntamente por dos universidades, Northwest University (China) y Lancaster (UK), reveló que, grabando con la cámara de un dispositivo móvil los movimientos realizados por el dedo del usuario al introducir su patrón de desbloqueo, y aplicando un algoritmo que extrae la geometría de dichos movimientos, es posible identificar un número muy reducido de patrones posibles (en muchos casos uno único) para el usuario. Es más, el estudio concluye que, a mayor complejidad del patrón, más sencillo resulta identificarlo con este tipo de ataque [Ref.- 276].
704. En las versiones anteriores a Android 6.x, en las que no existía el demonio "*gatekeeperd*", los patrones de bloqueo eran almacenados bajo `"/data/system"`, cifrados con un hash SHA-1 sin semilla (concretamente, en el fichero "*gesture.key*" para dispositivos con solo un usuario y bajo `"/data/system/users/<user_id>/gesture.key"` para dispositivos multi-usuario). Al no tener semilla, era posible calcular todas las posibles combinaciones del hash del patrón, por lo que disponer del fichero "*gesture.key*" (para acceder al cual es necesario disponer de acceso al dispositivo móvil como root) valdría para que un potencial atacante pudiera adivinar u obtener el gesto o patrón.
705. Desde Android 6.x, en aquellos dispositivos que disponen de un TEE (Trusted Execution Environment) el subsistema Gatekeeper participa en el proceso de autenticación en el dispositivo móvil, empleando el TEE.

706. El TEE es un área independiente que reside en el hardware del dispositivo. El TEE puede disponer de un procesador y una memoria dedicados o puede emplear una instancia virtual de la CPU principal. En cualquier caso, el TEE se encuentra completamente aislado del resto del sistema a través de lo que se conoce como la TrustZone (o "hardware-backed memory") y los mecanismos de entrada/salida hacia/desde él se encuentran protegidos, incluso en dispositivos sobre los que se ha realizado el proceso de *rooting*.
707. Para dar soporte al TEE, se define un conjunto de componentes software denominados "Trusty TEE" [Ref.- 622], que incluyen el sistema operativo que ejecuta en el procesador del TEE (Trusty OS), las apps que ejecutan dentro del TEE sobre el Trusty OS (apps de confianza o *trustlets*), los drivers del kernel de Android para que las aplicaciones que requieren de los servicios del TEE puedan comunicarse con él, y un conjunto de librerías encargadas de la comunicación entre los drivers del kernel y las apps de confianza.
708. En dispositivos móviles que disponen de TEE, en lugar de que el almacenamiento del patrón use un hash sin semilla como en versiones anteriores de Android o en aquellos dispositivos que no disponen de TEE, se emplea un HMAC (Hash-based Message Authentication Code) basado en una clave secreta que reside en el hardware del TEE del dispositivo. El TEE se encargará asimismo de calcular el HMAC durante el proceso de autenticación junto al Gatekeeper (ver apartado "11. Gestión de certificados digitales y credenciales") [Ref.- 275].
709. Este mecanismo, aunque no impide por completo ataques de fuerza bruta, sí los dificulta, pues el Gatekeeper puede rechazar el acceso tras un número determinado de intentos fallidos en un espacio de tiempo determinado.
710. "PIN": Permite establecer un PIN o código numérico tradicional, compuesto por al menos cuatro dígitos (0-9), no desvelándose en la pantalla de desbloqueo la longitud del mismo. Actualmente existe una limitación máxima de 16 dígitos para el PIN (si se excede, el terminal mostrará un mensaje que informa de que el PIN debe tener menos de 17 dígitos).



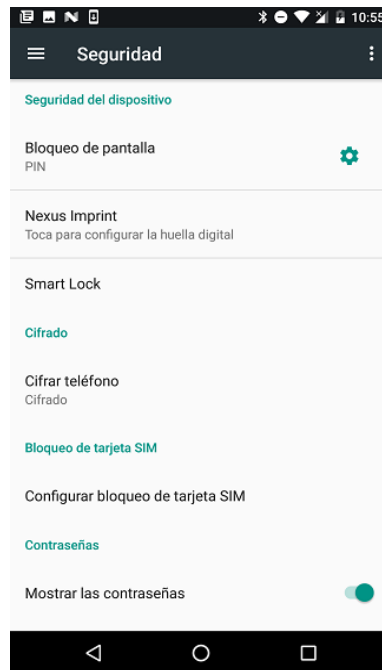
711.El método de PIN también hace uso de un hash almacenado, pero, a diferencia del patrón, se emplea una semilla aleatoria de 64 bits, que se almacena en la base de datos *locksettings.db*. Al igual que ocurría en el caso del patrón, en aquellos dispositivos móviles que incorporan un TEE, el proceso de autenticación y almacenaje de claves se lleva a cabo por parte del Gatekeeper.

712."Contraseña": permite establecer una contraseña alfanumérica, compuesta por un mínimo de cuatro y un máximo de 16^{67} caracteres alfanuméricos (incluyendo, si se desea, letras minúsculas, mayúsculas, dígitos y símbolos de puntuación) y por tanto, constituyéndose como la opción más segura y recomendada desde el punto de vista de seguridad. Si se excede la longitud máxima de la contraseña, de nuevo al igual que con el PIN, el terminal mostrará un mensaje indicando que el número de caracteres de la contraseña ha de ser inferior a 17 (el mensaje que notifica de esta última condición es confuso en el dispositivo móvil empleado en la elaboración de la presente guía, como ilustra la imagen inferior).



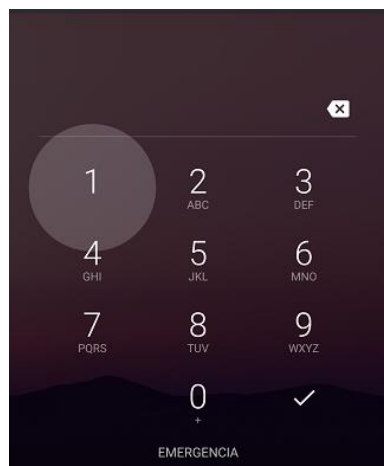
⁶⁷ <https://android.stackexchange.com/questions/154588/why-and-how-to-increase-16-character-lockscreen-password-limit>

713. Con la llegada de Android 4.0 (*Ice Cream Sandwich*), se incorporó una opción denominada "Desbloqueo facial" (*Face Unlock*), que permite desbloquear el dispositivo móvil al analizar mediante la cámara delantera del dispositivo el rostro del usuario y reconocer el patrón de rasgos biométricos configurado para el reconocimiento facial.
714. Con Android 5.x, este método de desbloqueo dejó de ser una opción en sí misma (dentro del apartado "Seguridad" de los "Ajustes" de Android) y se integró como parte de la funcionalidad *Smart Lock* (y pasó a denominarse *Trusted Face*, o "cara de confianza"). Así se mantiene en Android 7.x (ver apartado "9.5. Desbloqueo mediante Smart Lock").
715. "Huella dactilar digital" (o *Nexus Imprint*): desde Android 6.x (y para aquellos dispositivos que incorporan un lector de huellas), es posible desbloquear el dispositivo móvil mediante la huella dactilar digital del usuario (ver apartado "9.3. Huella dactilar digital (*fingerprint*)").
716. Pese a que los dispositivos Nexus 5X suministrados con Android 7.x de fábrica se encuentran cifrados (ver apartado "10. Cifrado de datos en Android"), y al igual que ocurría en Android 6.x, las opciones para el desbloqueo del dispositivo móvil "Ninguna" o "Deslizar", estarán disponibles (a diferencia de lo que ocurría en versiones previas de Android, como 4.x), debido a que pueden hacer uso de una contraseña por defecto para proteger la clave maestra de cifrado.
717. Las opciones "Sin seguridad" o "Deslizar" están completamente desaconsejadas desde el punto de vista de seguridad, ya que exponen el dispositivo móvil a accesos no autorizados si un potencial atacante obtiene acceso físico al mismo, incluso temporalmente.
718. Android no realiza ninguna verificación de seguridad sobre el valor del PIN o contraseña elegidos, permitiendo el uso de secuencias simples de dígitos o caracteres, como por ejemplo secuencias conocidas, que se repiten o que siguen un patrón entre sus dígitos o letras, como 1111, 1234, aaaa, abcd, etc.
719. Adicionalmente, en el caso de emplear una contraseña alfanumérica (opción recomendada), por defecto Android no impone restricciones complejas sobre su contenido; así, no obliga a que la contraseña conste de valores en diferentes categorías de las cuatro siguientes: letras minúsculas, letras mayúsculas, números (o dígitos) y símbolos de puntuación, opción que aporta la mayor seguridad.
720. Aunque en versiones anteriores de Android sí se imponía como única restricción para las contraseñas que contuviesen al menos una letra, para diferenciarlas de los PIN (basados únicamente en dígitos), en Android 7.x sí es posible definir una contraseña que solo contenga valores numéricos.
721. Al emplear una contraseña de acceso o un PIN, Android permite definir si se muestran o no los valores de los caracteres introducidos (en concreto, únicamente el último dígito o carácter pulsado) mientras se escribe en la pantalla de desbloqueo, a través de la opción "[Contraseñas] Mostrar las contraseñas" (habilitada por defecto) y disponible desde el menú "Ajustes [Personal] - Seguridad [Seguridad del dispositivo]".

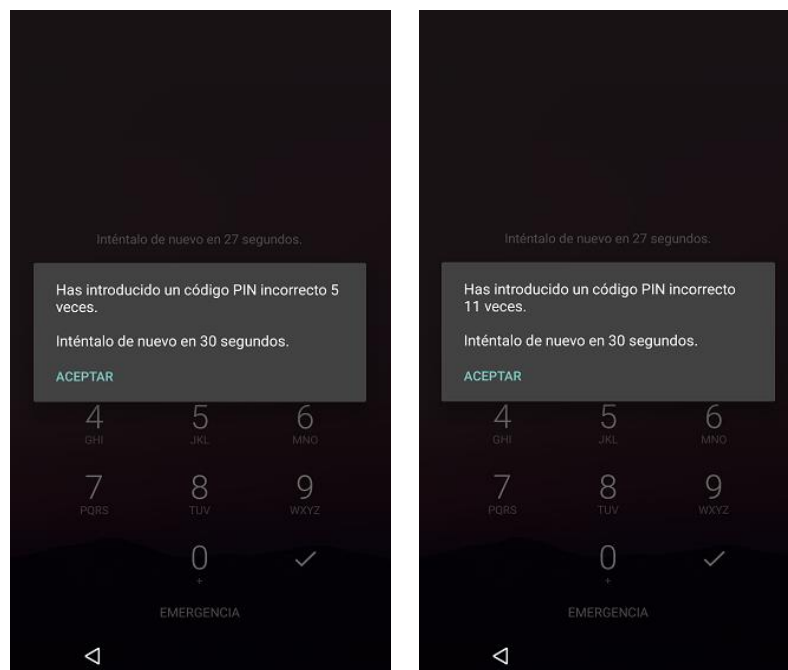


722.Desde el punto de vista de seguridad se recomienda no habilitar el uso de contraseñas visibles, para evitar que alguien con acceso visual temporal a la pantalla del dispositivo móvil durante el proceso de desbloqueo pueda llegar a leer total o parcialmente la contraseña empleada.

723.Lamentablemente, tanto en el caso del bloqueo con PIN como con contraseña, esta opción sólo implica que los caracteres (dígitos o letras) no se mostrarán en el espacio destinado a la introducción del código durante el propio proceso de configuración y selección del PIN o contraseña. Pero, aunque la opción de "Mostrar las contraseñas" esté desactivada, la pantalla de desbloqueo siempre muestra (mediante la iluminación de la tecla empleada y la representación correspondiente) el carácter pulsado por el usuario, aunque éste no se muestre en el campo de introducción del código. Por este motivo, entre otros, se debe prestar especial atención a quién dispone de acceso visual a la pantalla del dispositivo móvil durante el proceso de desbloqueo.



724. En base a las consideraciones anteriores, se considera mucho más seguro utilizar como código de acceso un PIN o una contraseña robustos frente al uso del patrón.
725. La introducción de un código de acceso basado en PIN, contraseña o patrón erróneo durante 5 intentos consecutivos provoca que Android bloquee durante 30 segundos la posibilidad de introducir de nuevo el código, con el objetivo de minimizar los ataques de fuerza bruta. Es importante reseñar que aquellos códigos erróneos que no cumplan los requisitos de un código válido (longitud mínima o máxima) no computan a efectos de número de intentos de código erróneo.
726. Como muestran las imágenes inferiores, tras introducir cinco veces un código de acceso incorrecto, el dispositivo móvil bloquea la posibilidad de acceso durante 30 segundos, notificando al usuario sobre este hecho y mostrando un contador con los segundos restantes hasta poder volver a intentar acceder al dispositivo móvil.
727. Si, transcurrido el plazo para poder volver a introducir el código de acceso, volviesen a realizarse 5 intentos de acceso fallidos, se volverá a establecer el plazo de 30 segundos en los que el dispositivo no permite la introducción del código.
728. Hasta Android 6.0.1 inclusive, el bloqueo del dispositivo durante 30 segundos tenía lugar cada 5 intentos fallidos. En Android 7.x, a partir del décimo intento fallido, cada nuevo intento erróneo provocará que el terminal se bloquee durante 30 segundos más, y en la pantalla se mostrará un mensaje que indica el total de intentos fallidos de introducción del código desde la última vez que el dispositivo se desbloqueó con éxito.



729. Este cambio provoca que se minimice la posibilidad de ataques de adivinación o fuerza bruta del código de acceso, que pasa de los 600 intentos por hora (considerando el caso ideal para el atacante, con un

máximo de cinco intentos cada 30 segundos, y despreciando el tiempo empleado para realizar los propios intentos de acceso) a 128 (10 en el primer minuto y $2 \cdot 59 = 118$ en los 59 minutos restantes).

730. Como regla general, es recomendable actualizar a la última versión del sistema operativo que esté disponible, pues a lo largo de la historia de Android ha habido vulnerabilidades que han afectado al método de desbloqueo y al código de acceso al dispositivo [Ref.- 145].

731. Para proceder a modificar el código de acceso del dispositivo móvil, en caso de disponer de un valor ya fijado, es necesario introducir el código de acceso (PIN, contraseña o patrón) actual. Este mecanismo de seguridad evita que un tercero con acceso temporal no autorizado al dispositivo móvil, en caso de encontrarse el dispositivo móvil desbloqueado, pueda modificar el código de acceso existente actualmente sin conocerlo.

732. La pantalla de desbloqueo, o *keyguard*, de Android ha permanecido sin novedades prácticamente desde Android 2.x hasta la versión 5.x, salvo por las capacidades multi-usuario añadidas en Android 4.x (tabletas) y 5.x (*smartphones*).

733. Sin embargo, mientras que en versiones anteriores a Android 4.x, las aplicaciones de terceros podían mostrar ventanas en la capa *keyguard*, por ejemplo para interceptar el botón Home en apps de tipo quiosco, a partir de Android 4.x la incorporación de ventanas a esta capa solo está permitida para aplicaciones de sistema (a través de un permiso de tipo *signature*) [Ref.- 159].

734. Desde Android 6.x, con el objetivo de habilitar las capacidades de auditoría de los códigos de acceso o contraseñas, el fichero "device_policies.xml" (disponible bajo "/data/system/" y solo accesible por root o system) almacena los detalles del formato establecido según la política de seguridad del dispositivo móvil para el código de acceso actual.

735. El formato del fichero es el siguiente, y desvela detalles como la longitud o el tipo de categorías permitidas para los caracteres que deben conformar la contraseña (letras mayúsculas o minúsculas, números, símbolos, etc.) [Ref.- 209]:

```
<policies setup-complete="true">
...
  <active-password length="6" letters="0" lowercase="0" nonletter="6"
                    numeric="6" quality="196608" symbols="0" uppercase="0">
  </active-password>
</policies>
```

736. Los detalles de este fichero podrían facilitar a un potencial atacante (que haya obtenido acceso como root en el dispositivo móvil) la realización de ataques de diccionario o fuerza bruta sobre el código de acceso o contraseña, al dar pistas a un potencial atacante que le permitan reducir el conjunto total de posibles combinaciones según la política de seguridad establecida.

737. Android 6.x introdujo nuevas modificaciones para permitir la autenticación o desbloqueo del dispositivo móvil mediante una huella dactilar digital, que

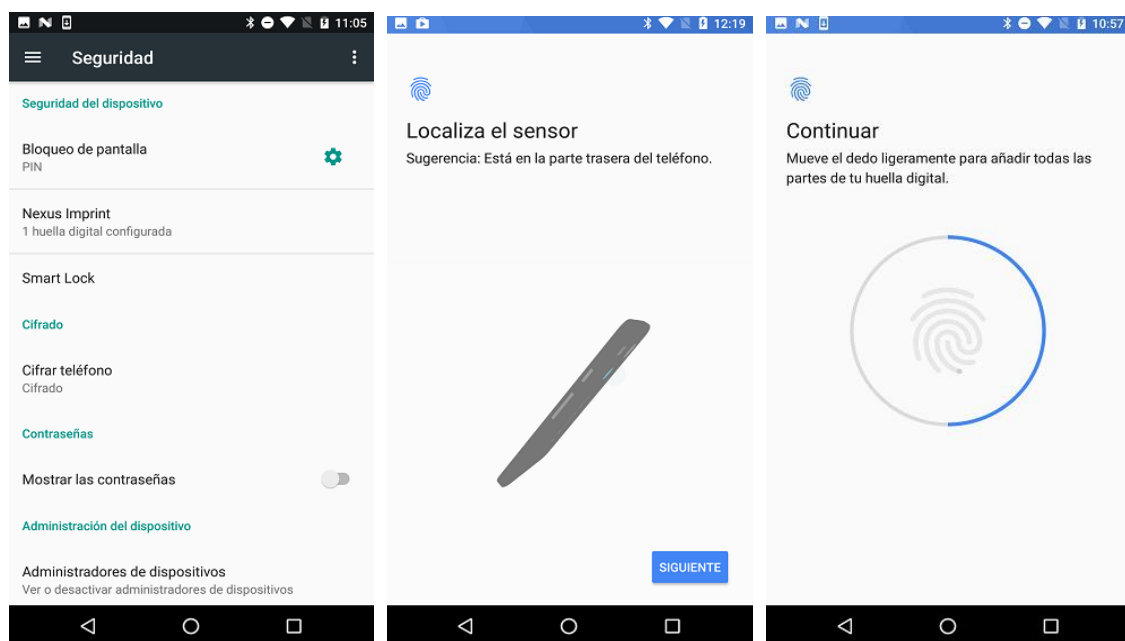
también afectan al almacenamiento del código de acceso y de otras contraseñas [Ref.- 620].

- 738.El "*gatekeeperd*" es un demonio de Android 6.x encargado de la gestión (registro y verificación) de los códigos de acceso del usuario. El código de acceso del usuario es convertido en una representación binaria del mismo (denominada *password handle*), y almacenado en forma de firma criptográfica, cuyo formato depende de la implementación del HAL (*Hardware Abstraction Layer*) del fabricante del dispositivo móvil, por ejemplo, haciéndose uso del TEE (*Trusted Execution Environment*) en el caso de la versión AOSP de Android mediante *scrypt* (como algoritmo de derivación de claves) con una semilla (o *salt*), y un identificador asociado a cada usuario [Ref.- 620].
739. El objetivo del Gatekeeper es almacenar de manera más robusta los diferentes códigos de acceso en Android, y con un nivel de seguridad similar, de manera que no sea más sencillo lanzar ataques de *cracking* (diccionario o fuerza bruta) sobre el patrón de desbloqueo (al no emplearse una semilla) que sobre un PIN o contraseña (tal como ocurría en la implementación de versiones previas de Android).

9.3. HUELLA DACTILAR DIGITAL (*FINGERPRINT*)

- 740.Desde Android 6.x se ofrece soporte nativo para llevar a cabo el desbloqueo de la pantalla a través de autenticación mediante una huella dactilar digital (o *fingerprint*). Estas capacidades únicamente están disponibles en aquellos dispositivos móviles que cuentan con un sensor de huella, como por ejemplo el Nexus 5X o 6P.
- 741.Existen en el mercado múltiples dispositivos móviles (*smartphones*) basados en Android 4.x y 5.x con sensor de huella dactilar digital, pero que no disponían de soporte nativo por parte de Android para su utilización (ni de APIs asociadas), por lo que hacían uso de una implementación propietaria de cada fabricante, dando lugar a debilidades de seguridad [Ref.- 610].
- 742.Algunos fabricantes de dispositivos móviles Android han diseñado implementaciones inseguras del sensor de huella dactilar digital, permitiendo tanto la interacción directa del kernel de Android (Linux) con el sensor hardware (y las librerías y servicios asociados), o incluso almacenando la representación de la huella dactilar digital del usuario en el sistema de ficheros sin cifrar [Ref.- 611]. Como resultado, una vez se dispone de acceso como root al dispositivo móvil, es posible tener acceso completo a la representación de la huella dactilar digital del usuario.
- 743.El sistema de autenticación por huella dactilar digital de los dispositivos móviles Nexus (y Pixel) se conoce como (Nexus) Imprint. La implementación de Imprint en los dispositivos móviles Nexus (y Pixel) obliga al almacenamiento y reconocimiento de la huella dactilar digital en el TEE (*Trusted Execution Environment*), con acceso hardware limitado y protegido por una política de SELinux.

744. Durante el registro de una huella dactilar digital, el denominado "Trusty OS" [Ref.- 622], que es el sistema operativo que gestiona el TEE, almacena un modelo o representación de la huella y un conjunto de datos adicionales para validarla, todos ellos cifrados y almacenados dentro del TEE.
745. El mecanismo de autenticación mediante huella dactilar digital puede ser más inseguro que el uso de un patrón, PIN y (especialmente) contraseña robustos, debido a que el usuario desvela (o distribuye) a lo largo del día y en numerosos lugares (incluyendo el propio dispositivo móvil) una copia o representación de su huella dactilar.
746. Por tanto, el mecanismo de autenticación mediante huella dactilar digital pretende ofrecer un equilibrio razonable entre la protección de la pantalla de desbloqueo del dispositivo móvil y la usabilidad, conveniencia y facilidad a la hora de tener que introducir el código de acceso decenas o centenares de veces al día por parte del usuario.
747. Como mecanismo de protección adicional, a la hora de configurar una huella dactilar digital es necesario seleccionar un mecanismo de autenticación de respaldo (patrón, PIN o contraseña), que será solicitado al usuario cada vez que se reinicia el dispositivo móvil, si no es posible reconocer su huella dactilar digital después de 5 intentos de acceso fallidos, al cambiar de usuario en el dispositivo móvil (en escenarios multi-usuario), y después de 48 horas sin haber desbloqueado el dispositivo móvil con el método de respaldo, añadiendo así múltiples factores de autenticación [Ref.- 609].
748. Es posible configurar una huella dactilar digital a través del menú "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Nexus Imprint". Al acceder a este menú es necesario introducir el código de acceso configurado actualmente en el dispositivo móvil.

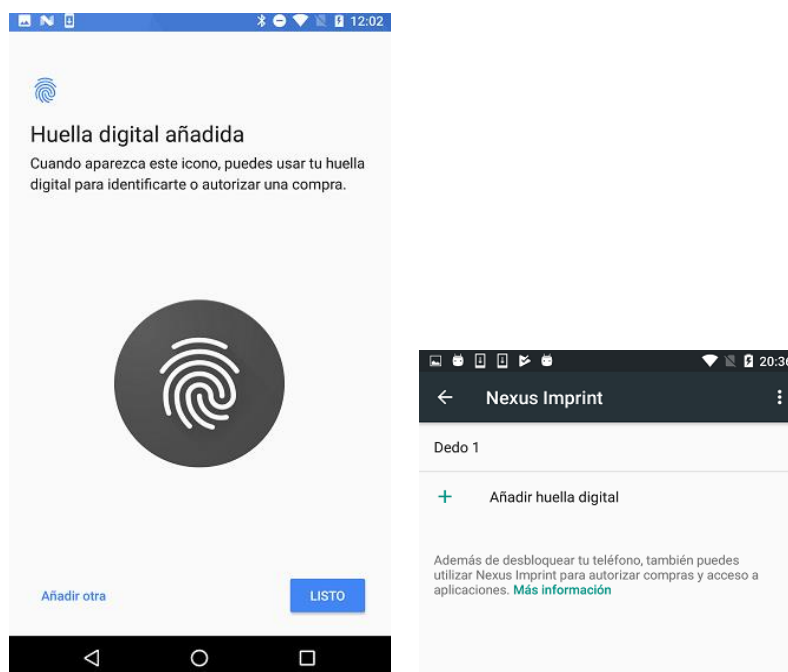


749. Desde dicho menú se muestra la pantalla de Nexus Imprint que indica dónde se localiza el sensor de huella dactilar digital (ubicado en la parte posterior del dispositivo móvil empleado para la elaboración de la presente guía). Una

vez localizada su ubicación, el botón "Siguiente" permite comenzar el proceso de reconocimiento de la huella dactilar del usuario, que muestra a través de un tutorial los pasos a seguir. Será necesario colocar y levantar el dedo en múltiples ocasiones (no menos de 6) sobre el sensor hasta que el dispositivo móvil confirme que dispone de los datos de representación de la huella y que esta se ha añadido como método de desbloqueo.

750. Es posible registrar hasta un máximo de 5 huellas dactilares digitales para desbloquear el dispositivo móvil, disponiendo de la posibilidad de renombrar cada una de ellas para su fácil identificación.

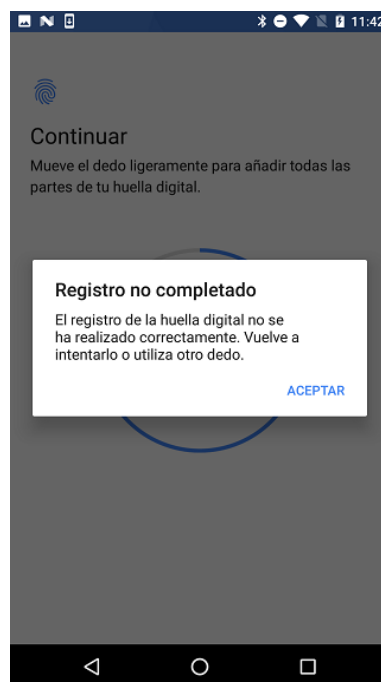
751. Como resultado, aparecerá una nueva entrada en la pantalla de Nexus Imprint, denominada por defecto "Dedo 1" (o "Finger 1"). Seleccionando la misma es posible cambiar el nombre que identifica a la representación de dicha huella, por ejemplo, para poder saber a qué usuario y dedo corresponde, o eliminar dicha huella dactilar digital del dispositivo.



752. Una vez registrada la primera huella dactilar digital, durante el proceso de registro inicial (o de una nueva huella posteriormente), se pueden añadir más huellas mediante el botón "Añadir huella digital". Una vez registradas todas las huellas dactilares deseadas durante un mismo proceso de registro, se puede finalizar la configuración mediante el botón "Listo".

753. Debido a que desde la pantalla de Nexus Imprint se pueden añadir, mediante la opción "Añadir huella digital", hasta un máximo de 5 huellas en total por dispositivo móvil, a la hora de registrar una nueva huella dactilar digital es necesario que se establezca una cadena de confianza mediante la autenticación del usuario, empleando el código de acceso de respaldo. Si transcurre un tiempo entre la introducción del código de acceso y la obtención de la pantalla de Nexus Imprint y el momento de inicio del proceso de registro y configuración de una nueva huella, se volverá a solicitar al usuario iniciar el proceso de registro, debiéndose introducir el código de acceso actual de nuevo.

754. En concreto, si transcurren más de 10 minutos entre el inicio del proceso para añadir una huella y la siguiente, o si se tarda ese tiempo desde que se accede al menú de configuración de Nexus Imprint y se inicia el proceso de registro y escaneo de una nueva huella, cuando se trate de reanudar dicho proceso, el sistema mostrará un mensaje de error con el título "Registro no completado".
755. Aunque hasta Android 6.x este mensaje se mostraba nada más reanudarse el proceso de configuración de la huella, en Android 7.x parecerá que el proceso puede retomarse, pero antes de su compleción, el sistema abortará y mostrará el mensaje de "Registro no completado". En este caso, será necesario iniciar el proceso de registro de nuevo, accediendo a la pantalla de Nexus Imprint y debiendo introducir de nuevo el código de acceso (PIN, patrón o contraseña) que esté activo como mecanismo de desbloqueo actualmente. Dicho código de acceso es empleado como mecanismo de respaldo al método de desbloqueo mediante huella dactilar digital, en caso de que no fuera posible autenticarse con la huella dactilar (por ejemplo, porque el usuario haya tenido un accidente con el dedo empleado para la autenticación y su huella dactilar se haya visto afectada).

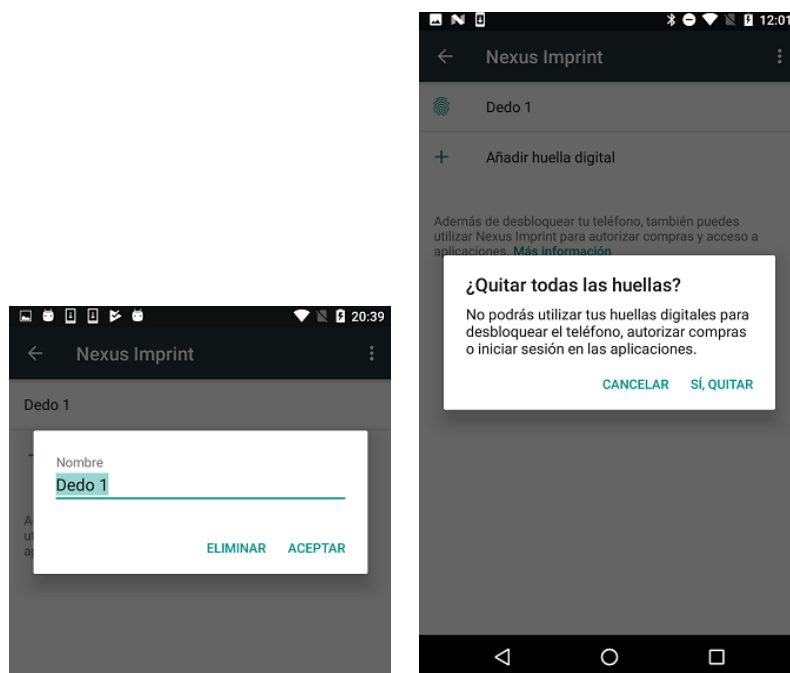


756. Este mecanismo de seguridad evita que un tercero con acceso temporal no autorizado al dispositivo móvil, en caso de encontrarse el dispositivo móvil desbloqueado, pueda registrar una nueva huella dactilar digital sin conocer el código de acceso existente actualmente.
757. El objetivo de proteger la representación (o imagen) de la huella dactilar digital por hardware, tal como ocurre en la implementación utilizada por el dispositivo móvil empleado para la elaboración de la presente guía, es evitar que pueda ser extraída del dispositivo móvil. Asimismo, tampoco esta representación es incluida en las copias de seguridad del dispositivo móvil, o compartida con los servidores de Google.

758. En caso de almacenarse esta representación en el sistema de ficheros (protección por software y opción posible en la implementación de otros modelos de dispositivo móvil Android), debe de hacerse en formato cifrado, adicionalmente al cifrado nativo de la memoria del dispositivo móvil. La representación también debe ser firmada criptográficamente empleando una clave privada específica del dispositivo móvil (y otros valores criptográficos asociados), con el objetivo de que no pueda ser reutilizada en otro dispositivo móvil o entre usuarios del mismo dispositivo móvil [Ref.- 608].
759. La disponibilidad por defecto de las capacidades de huella dactilar digital a partir de Android 6.x permitió estandarizar la arquitectura e implementación (incluyendo la guía de buenas prácticas) asociada al uso de este tipo de sensores en Android, mediante una Fingerprint Hardware Abstraction Layer (HAL) que enlaza el sensor hardware y la librería específica proporcionada por el fabricante, con las capacidades disponibles en Android [Ref.- 615].
760. El documento de definición de las reglas de compatibilidad de Google para Android 7.x ("Nougat"), "Android 7.0 (N) Compatibility Definition Document (CDD)", define ciertos prerrequisitos, como que deben limitarse los intentos de acceso al dispositivo móvil al menos durante 30 segundos tras 5 intentos fallidos de autenticación con la huella dactilar digital, que la representación de la huella dactilar digital del usuario debe protegerse en una Keystore implementada por hardware, que las verificaciones de la huella deben realizarse en el TEE (o un chip hardware dedicado) y que los datos asociados deben estar cifrados y firmados criptográficamente para que no puedan ser usados fuera del TEE, entre otros [Ref.- 606].
761. Cuando el método de acceso o desbloqueo por huella está activo, en lugar de un candado cerrado, la pantalla de bloqueo mostrará un icono de una huella dactilar en su parte central inferior.
762. La huella dactilar digital puede ser utilizada para desbloquear el dispositivo móvil, pero también se puede emplear para autorizar compras y pagos (mediante Android Pay y Google Play), o para autenticarse dentro de las apps. Cuando tal acción sea posible, el dispositivo móvil mostrará un mensaje solicitando la lectura de la huella dactilar, que en ningún caso se comparte con la aplicación. Es decir, las apps no disponen en ningún caso de acceso a la representación de la huella dactilar digital del usuario, sino que emplean la "Trusty API" [Ref.- 742] [Ref.- 743] para solicitar al sistema operativo la validez de la huella. El sistema operativo contactará con el TEE y, en función de la respuesta que reciba de él, proporcionará a la app un token de notificación de aceptación o de rechazo de la huella introducida por el usuario.
763. Para ello, Android 6.x añadió para los desarrolladores de apps nuevas APIs (FingerprintManager API) para que las apps puedan interactuar con el hardware asociado al sensor de huella dactilar digital. Estas deben de ser utilizadas e integradas conjuntamente con el Keystore de Android (ver apartado "11.1. Android Keystore System: Almacenamiento de credenciales"), junto a un nuevo permiso que debe ser declarado por las

apps: USE_FINGERPRINT. Las apps pueden integrar las capacidades de autenticación mediante huella dactilar digital con otras nuevas funcionalidades de Android 6.x, como las APIs de pagos móviles (Payment APIs) [Ref.- 603].

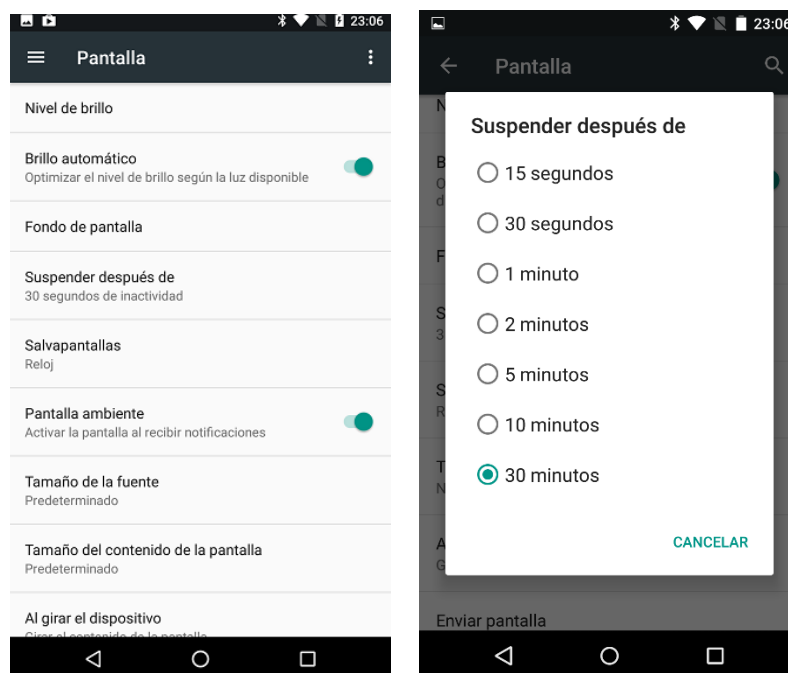
764. Adicionalmente, las nuevas APIs de confirmación de credenciales (Confirm Device Credential APIs) permiten a las apps gestionar claves o tokens criptográficos (pudiendo hacer uso de una implementación mediante clave pública o secreta), protegidos por hardware en el Keystore, asociarlos a la pantalla de desbloqueo, y definir políticas de acceso a los mimos (por ejemplo, si el usuario se ha autenticado en el dispositivo móvil en los últimos 5 minutos) [Ref.- 607].
765. De esta forma la app puede autenticar al usuario mediante la autenticación o el desbloqueo del propio dispositivo móvil (en lugar de disponer de un mecanismo de autenticación adicional propio), o conocer si este ha sido desbloqueado recientemente, pudiendo emplear referencias de tiempo de cuándo fue la última vez que se desbloqueó el mismo [Ref.- 603].
766. Esta funcionalidad evita que el usuario tenga que recordar una contraseña adicional para el acceso a la propia app, y elimina posibles errores a la hora de que las apps implementen su propio mecanismo de autenticación y el interfaz de usuario asociado.
767. Si en los ajustes de configuración de la huella dactilar digital se muestra el mensaje "*Screen lock option disabled*", indica que el administrador del dispositivo móvil ha establecido una política de seguridad que no permite el uso de una huella dactilar digital, obligando al usuario a utilizar un patrón, PIN o contraseña (por ejemplo, en la cuenta de trabajo de Android for Work; ver apartado "8.2. Android for Work"). Con esta configuración, todavía es posible hacer uso de una huella dactilar digital para autorizar compras o autenticarse en apps.
768. En Android 7.x, la configuración y almacenamiento de la huella dactilar se gestiona independientemente para cada usuario. Si la política del dispositivo (DPC, Device Policy Client) utiliza la API de nivel 23 (o inferior) en un dispositivo con Android 7.x, el usuario podrá registrar una huella en el dispositivo móvil, pero no acceder a aplicaciones de trabajo. Con la API de nivel 24 y superiores, el usuario puede registrar la huella específicamente para el perfil de trabajo accediendo al menú "Ajustes - Seguridad - Seguridad del perfil de trabajo" [Ref.- 703].
769. Si después de configurar este método de acceso se desea dejar de hacer uso del desbloqueo con huella, es necesario acceder a "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Nexus Imprint" y seleccionar una por una las huellas añadidas, pulsando sobre su alias y en la opción "Eliminar/Quitar". Con la última huella eliminada se mostrará un mensaje que informa de que no se podrá seguir haciendo uso de las huellas como método de desbloqueo, pago o inicio de sesión en aplicaciones.



9.4. BLOQUEO Y DESBLOQUEO DE PANTALLA

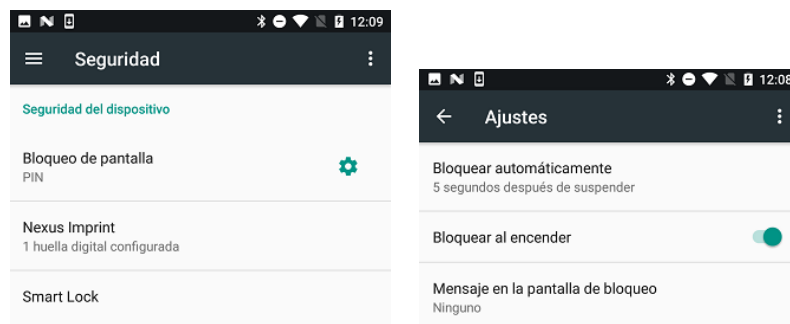
- 770.El término bloqueo de pantalla refleja el escenario en el que un dispositivo móvil activa la protección de la pantalla mediante un código de acceso.
- 771.El término desbloqueo de pantalla refleja el escenario en el que un dispositivo móvil requiere que el usuario haga uso del código de acceso definido actualmente en el dispositivo móvil para poder acceder al mismo, a sus datos y aplicaciones. A lo largo de la historia de Android han existido diversas vulnerabilidades asociadas a la pantalla de bloqueo [Ref.- 77]. Dado que el mecanismo de bloqueo es la llave de acceso al dispositivo, se recomienda prestar especial atención a todos los elementos que afectan al mismo.
- 772.El bloqueo de la pantalla del dispositivo móvil puede ser manual o automático. El bloqueo manual tiene lugar cuando el usuario pulsa el botón de encendido (no de manera prolongada, para evitar apagarlo). El bloqueo de pantalla es inmediato tras pulsar este botón. El bloqueo automático se produce al presentarse las condiciones de bloqueo establecidas a través de los diversos ajustes de configuración del dispositivo móvil.
- 773.Adicionalmente, la pantalla estará bloqueada inicialmente al arrancar el dispositivo móvil.
- 774.Para desbloquear la pantalla activamente, el usuario ha de pulsar en el botón de encendido y emplear el código de acceso definido actualmente en el menú "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Bloqueo de pantalla" (ver apartado "9.2. Código de acceso al dispositivo móvil"). A partir de Android 6.x, en caso de haber registrado una huella dactilar digital a través de la pantalla de configuración de Nexus Imprint, este método también permitirá desbloquear la pantalla (ver apartado "9.3. Huella dactilar digital (*fingerprint*)").

775. El bloqueo automático de pantalla tras un periodo de inactividad está definido por dos factores o parámetros de configuración (analizados en detalle a continuación): por un lado, el temporizador definido para suspender la pantalla tras un periodo de inactividad y, por otro, el temporizador definido para bloquear la pantalla después de haber sido suspendida.
776. Se recomienda por tanto establecer el bloqueo del dispositivo móvil tras un tiempo de inactividad, mediante el ajuste de la configuración de ambos parámetros, por ejemplo, a 1 minuto. El rango razonable, buscando el equilibrio entre seguridad y usabilidad, debería estar entre 1 y 2 minutos, pero debe ser definido por la política de seguridad de la organización propietaria del dispositivo móvil.
777. El temporizador de suspensión de la pantalla por inactividad establece qué tiempo ha de transcurrir desde que el usuario llevó a cabo la última interacción con el dispositivo móvil hasta que la pantalla entre en modo de ahorro de energía y se apague. Este parámetro se define a través del menú "Ajustes [Dispositivo] - Pantalla - Suspender después de":



778. Android proporciona un rango de valores temporales para suspender la pantalla de entre 15 segundos y 30 minutos, siendo el valor por defecto de 30 segundos.
779. Como se puede ver, Android no dispone de una opción que permita que la pantalla permanezca indefinidamente activa y que no se suspenda en ningún momento, independientemente del periodo de inactividad, con el objetivo de gestionar el consumo de batería del dispositivo móvil, siendo el periodo máximo de 30 minutos.
780. Por otro lado, el temporizador de bloqueo de la pantalla una vez ésta se ha suspendido por inactividad se establece a través del menú "Ajustes [Personal] - Seguridad [Seguridad del dispositivo] - <Rueda de ajustes junto

al tipo de bloqueo> - Bloquear automáticamente" (este menú no se muestra si, el método de desbloqueo es "Ninguno" o "Deslizar", es decir, si no se ha definido un código de acceso).

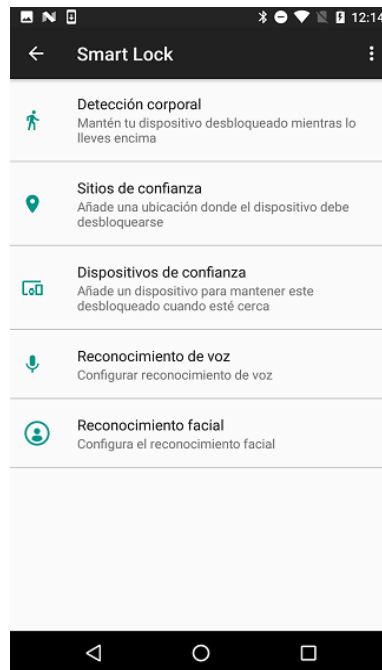


781. Android proporciona un rango de valores temporales para bloquear la pantalla tras suspender el dispositivo móvil de entre 5 segundos y un máximo de 30 minutos, con la posibilidad de bloquear la pantalla inmediatamente (opción más segura), siendo el valor por defecto (una vez fijado un código de acceso) de 5 segundos.
782. Este bloqueo automático no se aplicará si se cumplen algunas de las condiciones asociadas a la funcionalidad Smart Lock (ver apartado "9.5. Desbloqueo mediante Smart Lock"), lo que mantendrá el dispositivo móvil desbloqueado (por ejemplo, al estar en un sitio de confianza, o identificar un dispositivo de confianza).
783. Como se puede ver, Android no dispone de una opción que permita que la pantalla quede sin bloquear indefinidamente tras un periodo de inactividad, opción desaconsejada desde el punto de vista de seguridad en el caso de que existiera, siendo el periodo máximo de 30 minutos.
784. En resumen, se recomienda encontrar el equilibrio entre ambos parámetros de configuración para que se lleve a cabo la activación de la pantalla de desbloqueo tras un periodo de inactividad, por ejemplo, fijando el valor de "Suspender después de" a 1 minuto y el de "Bloquear automáticamente" a 30 segundos, lo que implicaría que pasarían menos de 2 minutos antes del bloqueo automático.
785. Adicionalmente, la configuración asociada al bloqueo de la pantalla del dispositivo móvil tras un periodo de inactividad se complementa con el siguiente parámetro de configuración, que determina si la pantalla debe ser bloqueada cuando la pantalla del dispositivo móvil es apagada manualmente por el usuario mediante el botón de encendido (pero únicamente la pantalla, sin apagar por completo el dispositivo móvil).
786. La opción "Bloquear al encender", disponible desde el menú "Ajustes [Personal] - Seguridad [Seguridad del dispositivo]", cuando está activa, implica que el evento de apagar la pantalla manualmente por parte del usuario (mediante el botón de encendido) forzará un bloqueo de la pantalla. Al encenderla de nuevo será necesario siempre introducir el código de acceso para desbloquearla. Es, por tanto, muy importante desde el punto de vista de seguridad el activar esta opción para posibilitar el bloqueo de pantalla manual rápidamente y en cualquier momento.

- 787.Sin embargo, si esta opción no es activada, el evento de apagar la pantalla manualmente por parte del usuario es equivalente a que el dispositivo móvil se suspenda tras un periodo de inactividad. Dependiendo del valor del parámetro "Bloquear automáticamente" (descrito previamente) se bloqueará la pantalla tras el tiempo especificado.
- 788.Aunque no se especifica en la opción "Bloquear al encender", el bloqueo al encender no se aplicará si se cumple alguna condición de la funcionalidad Smart Lock (ver apartado "9.5. Desbloqueo mediante Smart Lock"), lo que mantendrá el dispositivo móvil desbloqueado. Este comportamiento sí se indicaba mediante un mensaje descriptivo en versiones anteriores de Android, como Android 5.x y 6.x.

9.5. DESBLOQUEO MEDIANTE SMART LOCK

- 789.Smart Lock de Google es una de las principales novedades introducidas en Android 5.x ("Lollipop"). Se trata de una funcionalidad que permite que el dispositivo móvil se mantenga desbloqueado en determinadas circunstancias (por ejemplo, si el dispositivo móvil está en una ubicación configurada por el usuario, o si está conectado a un dispositivo Bluetooth considerando de confianza), a pesar de que se cumplan otras condiciones de bloqueo, por ejemplo, por tiempo de inactividad. Además, Smart Lock permite también el desbloqueo "automático" del dispositivo móvil, omitiendo el resto de mecanismos de seguridad existentes, como el código de acceso, huella dactilar digital, etc., bajo ciertas circunstancias como, por ejemplo, si detecta la cara o la voz registradas por el usuario (en concreto, por el usuario considerado por Android el propietario del dispositivo, en dispositivos multi-usuario).
- 790.Smart Lock se basa en un concepto denominado "agentes de confianza" (*Trust Agents*) que se introdujo en Android 5.x. Un agente de confianza es un servicio que, de acuerdo a la documentación del *framework* de Smart Lock, notifica al sistema sobre si considera que el entorno del dispositivo móvil es "de confianza" según sus propios criterios de evaluación. En caso de encontrarse en un entorno "de confianza", el sistema decide cómo relajar la seguridad del dispositivo, por ejemplo, no bloqueando la pantalla [Ref.- 206].
- 791.Para poder utilizar la funcionalidad de Smart Lock, es necesario acceder al menú "Ajustes - [Personal] Seguridad - [Ajustes avanzados] Agentes de confianza - Smart Lock (Google)".
- 792.Al activar Smart Lock como agente de confianza, se activará el menú "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Smart Lock", y bajo él se encuentran todas las opciones de configuración de Smart Lock y los distintos métodos de evaluación disponibles. En adelante, este acceso a través de los ajustes será abreviado como menú "Smart Lock".



793. Al pulsar sobre Smart Lock, siempre se solicitará al usuario el código de acceso al dispositivo móvil. Si no se ha definido ninguno hasta el momento, el dispositivo solicitará que se configure uno antes de poder acceder al menú Smart Lock.

794. Si el dispositivo no se utiliza durante 4 horas o es reiniciado, será necesario desbloquearlo manualmente con el código de acceso establecido actualmente para que las opciones de Smart Lock vuelvan a estar disponibles y operativas.

795. Las opciones de desbloqueo asociadas a Smart Lock, cada una de ellas configurable individualmente, son: (descritas en los siguientes puntos) [Ref.-206]

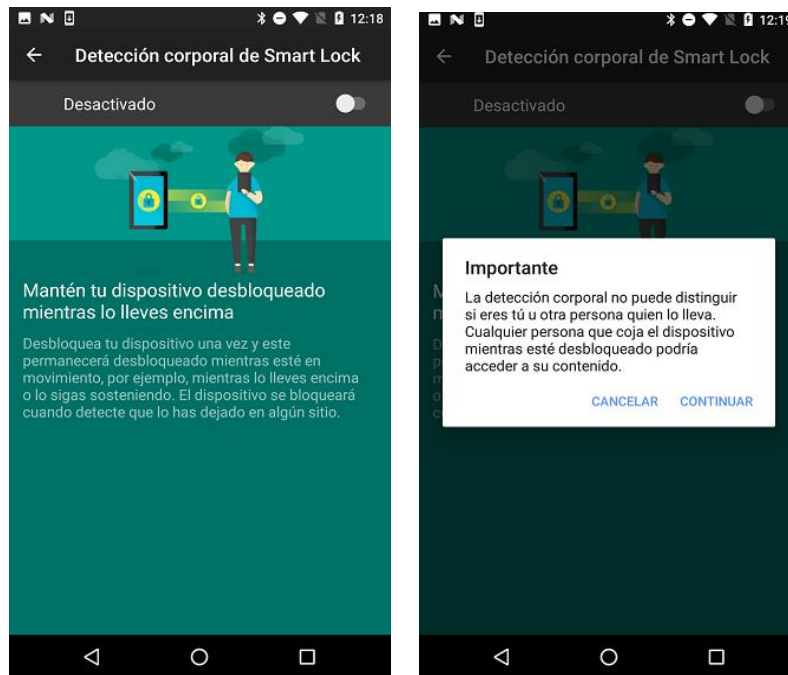
- Detección corporal.
- Sitios de confianza.
- Dispositivos de confianza.
- Reconocimiento facial.
- Reconocimiento de voz.

796. Es importante reseñar que, si se da una situación temporal en las que se requiera que el mecanismo de bloqueo clásico de Android siga activo, aunque se cumpla alguna de las condiciones asociadas a Smart Lock, no será necesario acceder al menú Smart Lock, sino que bastará con pulsar momentáneamente el candado de desbloqueo, con lo que se mostrará el mensaje "El dispositivo permanecerá bloqueado hasta que se desbloquee manualmente".

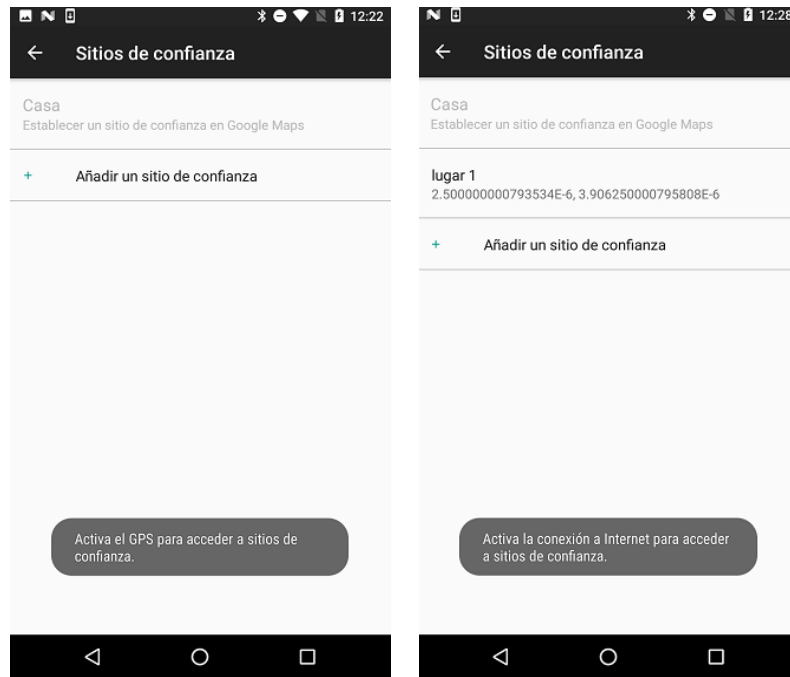
797. "**Detección corporal**": permite mantener el dispositivo móvil desbloqueado cuando el usuario lo lleva encima. Este método está disponible para dispositivos móviles que cuentan con un acelerómetro integrado, de forma

que el terminal se mantiene desbloqueado mientras el acelerómetro detecta que aún está en movimiento. En algunos dispositivos móviles, la detección corporal reconoce incluso el patrón que el propietario del dispositivo utiliza (o genera) al caminar. Si el acelerómetro detecta un patrón de desplazamiento que es significativamente diferente al almacenado, es posible que el dispositivo se bloquee. Es importante reseñar que la detección corporal no puede detectar quién tiene el dispositivo móvil encima, por lo que, si un tercero no autorizado dispusiese de acceso al dispositivo móvil mientras éste está desbloqueado, podría conseguir que permaneciese desbloqueado debido a esta función.

798. Adicionalmente, existen otros escenarios de riesgo, como cuando el usuario deja el dispositivo móvil sin bloquear en algún sitio, por ejemplo, sobre una mesa, si el mismo está configurado para bloquearse tras un tiempo (mediante el vencimiento del temporizador de bloqueo automático). Durante ese tiempo el dispositivo móvil aún estará expuesto a ser accedido por un potencial atacante, que podrá mantenerlo desbloqueado mediante la detección corporal.
799. Por este, y muchos otros motivos y escenarios detallados posteriormente, los métodos de Smart Lock, como la detección corporal, son significativamente menos seguros que un código de acceso (PIN, patrón o contraseña), por lo que se desaconseja su utilización.
800. Otro problema inherente a la detección corporal es el comportamiento del acelerómetro en los medios de transporte. Al utilizar un coche, un autobús, un tren u otro vehículo, el dispositivo móvil puede tardar entre cinco y diez minutos en bloquearse, al seguir detectando movimiento. El desplazamiento en avión o en barco puede provocar que el dispositivo no se bloquee automáticamente en ningún momento. Ambos factores han de tenerse en cuenta por si se considera oportuno bloquear el dispositivo móvil manualmente si fuese necesario.
801. Los datos del acelerómetro sobre el patrón de desplazamiento del usuario se almacenan en el dispositivo móvil, pero se eliminan cuando la función de detección corporal de Smart Lock es desactivada.
802. Para activar la función de Smart Lock de detección corporal, seleccionar el menú Smart Lock, se solicitará el código de acceso al dispositivo, y la opción "Detección corporal", cambiando el interruptor "Desactivado" a la posición de "Activado". El sistema solicitará confirmación, informando al usuario de que este método no puede determinar quién es el que porta el dispositivo:



803. **"Sitios de confianza"**: permite mantener el dispositivo móvil desbloqueado cuando el dispositivo detecta que se encuentra en una ubicación considerada de confianza, que ha debido ser previamente configurada, por ejemplo, el domicilio o el lugar de trabajo del usuario. Este método está disponible únicamente para dispositivos móviles que cuentan con un módulo GPS integrado y requiere que la ubicación esté activada. Hay que reseñar que el dispositivo no se desbloqueará al entrar en el radio de cercanía del sitio de confianza, sino que únicamente se mantendrá desbloqueado si ya se encontraba en una ubicación de confianza, y permanece dentro de ésta.
804. Dado que el servicio de ubicación no es cien por cien preciso, la función sitios de confianza puede mantener el dispositivo desbloqueado en un radio de hasta 80 metros. Por esta razón, desde el punto de vista de seguridad, se desaconseja hacer uso de esta funcionalidad especialmente en lugares públicos, como el centro de trabajo, cafetería, restaurante, biblioteca, gimnasio, etc., pues el dispositivo móvil podría quedar accesible para un tercero no autorizado.
805. Otro aspecto importante desde el punto de vista de seguridad es que las señales de ubicación o localización se pueden copiar y/o manipular, por lo que un hipotético atacante podría utilizar equipos especializados para mantener desbloqueado el dispositivo móvil en lugares que no son de confianza mediante la falsificación o suplantación de la señal GPS.
806. Para configurar los sitios de confianza en Smart Lock, hay que acceder al menú Smart Lock, introducir el código de acceso al dispositivo, y la opción "Sitios de confianza - Añadir un sitio de confianza". Si el servicio de ubicación no está activo, se informará al usuario de que es necesario activarlo mediante un mensaje Toast, "Activa el GPS para acceder a sitios de confianza":



807. Adicionalmente, para activar esta funcionalidad, el dispositivo requiere disponer de conexión a Internet, a fin de poder descargar la información de mapas para identificar el sitio a añadir.

808. No obstante, para el registro de un nuevo sitio de confianza no es necesario tener la ubicación activada, ya que se permite al usuario buscar su dirección de interés desplazándose manualmente por el mapa mostrado, o introduciendo los detalles de la dirección en el campo de búsqueda de la parte superior.

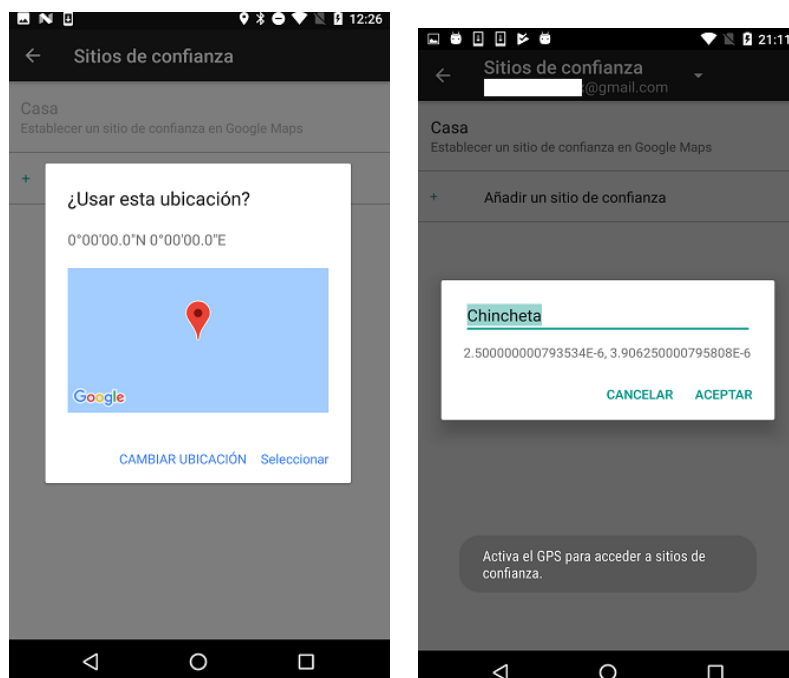


809. Por tanto, la activación del servicio de ubicación podría posponerse hasta que realmente se haya configurado adecuadamente el dispositivo móvil.

810. Tras disponer de los requisitos necesarios, aparece el menú "Elegir un sitio", que muestra un mapa de "Google Maps", con la ubicación actual si el servicio de ubicación está activo (y sobre el Ecuador si no lo está).

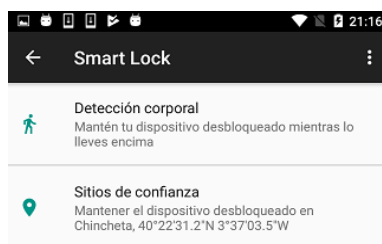


811. Si se desea utilizar la ubicación que se muestra en el mapa como sitio de confianza (que lógicamente requiere que el servicio de ubicación esté activo), se podrá pulsar la opción "Seleccionar esta ubicación". Si no, se puede pulsar en la lupa que aparece en la esquina superior izquierda del dispositivo móvil, asociada al campo de búsqueda disponible sobre el mapa, para llevar a cabo la búsqueda del sitio de confianza deseado en base a su nombre y/o dirección. La búsqueda puede realizarse mediante el nombre de una dirección o mediante coordenadas GPS (latitud y longitud). También es posible desplazarse tocando sobre el mapa hasta llegar a la ubicación deseada.
812. Tras pulsar en "Seleccionar esta ubicación", aparecerá una ventana de confirmación "¿Usar esta ubicación?", que permite "Seleccionar" o "Cambiar ubicación". Si se pulsa en "Seleccionar", aparecerá una ventana que permitirá dar nombre a la ubicación (por defecto, muestra las coordenadas de la misma):



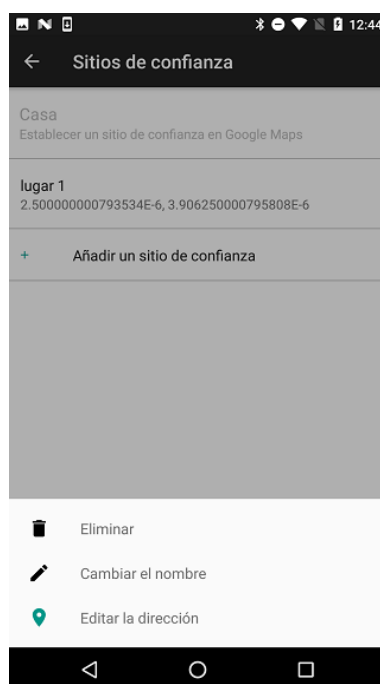
813. Es importante reseñar que, si se ha configurado una cuenta de usuario de Google en el dispositivo móvil (ver apartado "Cuenta de usuario de Google" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]), los sitios de confianza serán asociados a la cuenta, por lo que estarán disponibles de forma global en todos los dispositivos vinculados a ella (este aspecto se ilustra en la imagen superior derecha).

814. Al pulsar sobre el lugar elegido, aparecerá la ventana que permite confirmar la ubicación seleccionada, y tras confirmar su registro, se muestra la ventana de edición del sitio de confianza, que permite establecer un nombre para el marcador asociado (por defecto, el nombre puede incluir las coordenadas del lugar: latitud y longitud). Ésta será la referencia que aparecerá en el listado de sitios asociado al menú Smart Lock dentro de la opción "Sitios de confianza - <marcador>".

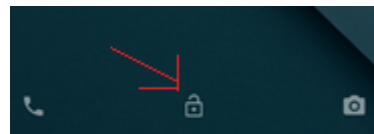


815. Por defecto, el menú sitios de confianza presenta un marcador llamado "Casa". Este marcador estará sombreado (y no será configurable) si el dispositivo no se ha vinculado a una cuenta de usuario de Google. En caso contrario, al pulsar sobre él, se podrá escribir la dirección particular o acceder a ella a través del mapa. La selección del sitio mediante el mapa hace uso de "Google Maps". Es importante reseñar, tal como informa el mensaje que se muestra en la parte inferior del dispositivo, que la dirección particular se actualizará en todos los demás productos de Google que la utilicen (por ejemplo, Búsqueda de Google y el propio Google Maps).

816. Opcionalmente, en algunas versiones previas de Android (por ejemplo, 5.0.x), también aparecía por defecto el marcador para definir la ubicación del "Trabajo" (además de "Casa").
817. Para eliminar un sitio de confianza, se ha de pulsar brevemente sobre el marcador del sitio que se muestra en el listado del menú "Smart Lock - Sitios de confianza". Como resultado aparecerá un menú en la parte inferior de la pantalla con varias opciones, como cambiar el nombre, editar la dirección o eliminar el sitio.

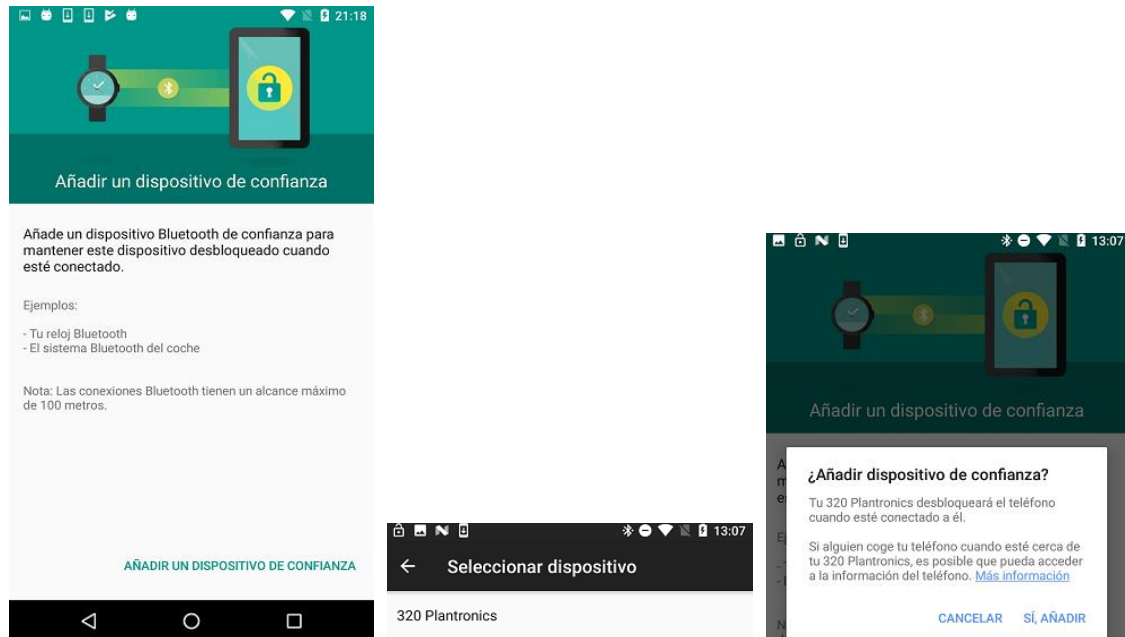


818. De manera general, desde el punto de vista de seguridad, se desaconseja el uso de la funcionalidad "Sitios de confianza" de Smart Lock.
819. **"Dispositivos de confianza"**: mediante esta función, es posible que el dispositivo móvil Android se mantenga desbloqueado cuando esté conectado con otro dispositivo (de confianza) mediante Bluetooth.
820. Hasta Android 6.x inclusive, se ofrecía la posibilidad de definir como dispositivo de confianza un dispositivo conectado a través de NFC. Sin embargo, Google decidió eliminar el soporte a NFC para la funcionalidad "dispositivos de confianza" en Android 7.x [Ref.- 744].
821. Hay que reseñar que el dispositivo móvil no se desbloqueará cuando detecte un dispositivo de confianza, únicamente se mantendrá desbloqueado en presencia de dicho dispositivo si ya se encontraba en ese estado. Así, aunque el usuario pulse el botón lateral de bloqueo, aunque esté configurado para bloquearse mediante el mismo, el dispositivo no se bloqueará si está conectado a un dispositivo de confianza.
822. Si se desea bloquear el dispositivo aun en presencia de un dispositivo de confianza, se deberá pulsar el botón de apagar pantalla y realizar una pulsación corta sobre el candado situado en el margen inferior:



823. No se recomienda en ningún caso utilizar como dispositivos de confianza elementos que siempre acompañen al dispositivo móvil, como por ejemplo las fundas o teclados Bluetooth. Tampoco se recomienda confiar en dispositivos que puedan ser sustraídos a la vez que el dispositivo móvil, como, por ejemplo, altavoces Bluetooth portátiles que se transporten en un mismo maletín. Desde el punto de vista de seguridad, tampoco se recomienda establecer esta relación de confianza con un reloj Bluetooth (o cualquier otro dispositivo de tipo *wearable*), ya que ambos podrían ser sustraídos simultáneamente al propietario (por la fuerza o por un descuido), sin que éste dispusiera de margen para bloquearlos.
824. Para configurar un dispositivo de confianza Bluetooth, hay que activar el interfaz Bluetooth en los ajustes del dispositivo móvil y, en primer lugar, establecer un emparejamiento con el dispositivo en el que se va a confiar. Este requisito, y su correcta verificación a la hora de establecerse una conexión Bluetooth entre ambos dispositivos, constituyen una mejora introducida en Android 5.1 (junto a Google Play Services, en abril de 2015), dado que, en versiones previas de Android 5.x con Smart Lock, un atacante podía simplemente suplantar la dirección Bluetooth del dispositivo de confianza y, mediante una conexión Bluetooth no cifrada, conseguir que el dispositivo móvil permaneciese desbloqueado [Ref.- 517].
825. Una vez emparejado por Bluetooth el dispositivo de confianza, es necesario acceder al menú "Smart Lock (se solicitará el código de acceso al dispositivo) - Dispositivos de confianza - Añadir un dispositivo de confianza - Bluetooth" (debiendo estar el interfaz Bluetooth activo; ver apartado "15. Comunicaciones Bluetooth").
826. Como resultado aparecerá la lista de dispositivos Bluetooth emparejados actualmente, de entre los cuales se podrá seleccionar el deseado. Finalmente, se mostrará una pantalla informativa para añadir el dispositivo

de confianza seleccionado y se solicitará confirmación por parte del usuario mediante la opción "Sí, añadir", o "Cancelar".



827. Cuando se realiza un emparejamiento con un dispositivo Bluetooth, Smart Lock generará una notificación invitando al usuario a configurar dicho dispositivo como dispositivo de confianza. Si se abre la notificación, se mostrará el menú que permite añadir dicho dispositivo a Smart Lock.

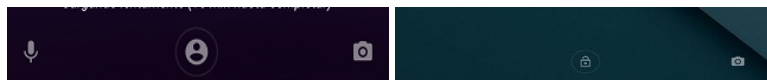
828. **"Reconocimiento facial"**: este método se introdujo en Android 4.0 (Ice Cream Sandwich) bajo el término "Desbloqueo facial" (Face Unlock), y dejó de ser una opción por sí misma en los ajustes de configuración para integrarse como parte de Smart Lock desde Android 5.x (siendo renombrado a Trusted face, o cara de confianza). Este método permite emplear biometría, y en concreto, los rasgos de la cara del usuario para desbloquear el dispositivo móvil mediante reconocimiento facial, tomando previamente, durante el proceso de configuración, una imagen de la cara del usuario.

829. Asimismo, la utilización de la opción de "Desbloqueo facial" está desaconsejada desde el punto de vista de seguridad, ya que es posible desbloquear el dispositivo móvil empleando una fotografía o imagen del usuario.

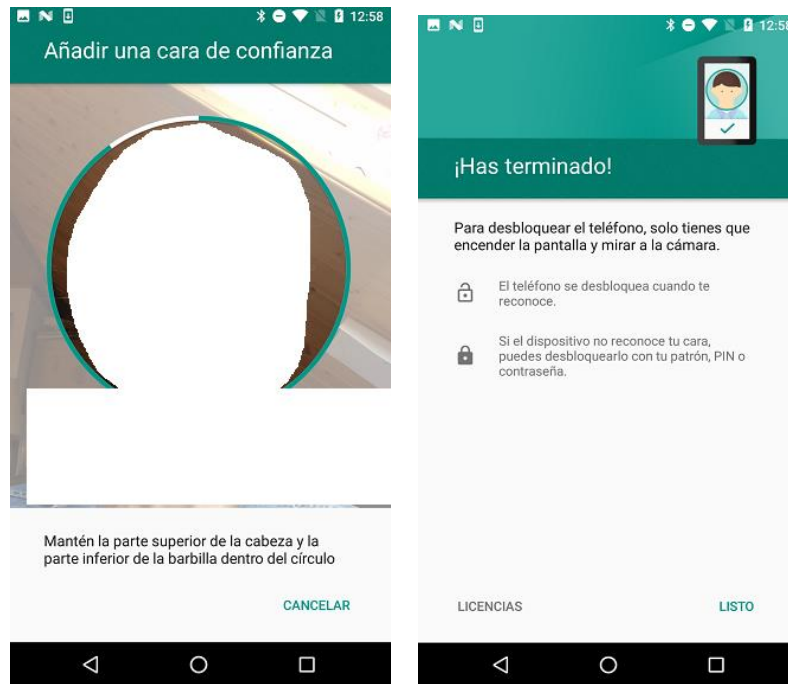
830. Para mitigar ese riesgo del método de desbloqueo facial, en versiones previas de Android (en concreto, disponible desde Android 4.1, *Jelly Bean*) se añadieron opciones como por ejemplo "Reconocimiento facial (Solicitar parpadeo al desbloquear)", para verificar que no se está empleando una fotografía estática. Sin embargo, esta verificación también puede ser potencialmente evitada empleando imágenes animadas (GIF) o vídeos. Desde Android 6.x ya no se dispone de esta opción.

831. Adicionalmente, Android 4.1 también introdujo la opción "Mejorar el reconocimiento facial", que permite capturar la cara del usuario en distintos escenarios (distintas iluminaciones, con o sin gafas, afeitado y sin afeitado, etc), con el objetivo de obtener los rasgos principales de la misma.

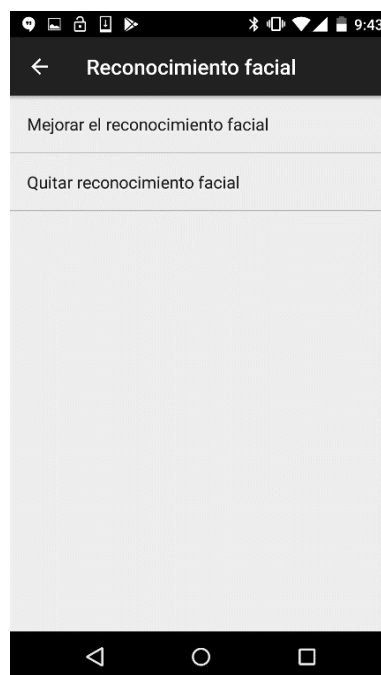
832. En Android 7.x, el método de desbloqueo mediante el rostro se configura con la opción "Reconocimiento facial", y requiere que se fije un método de desbloqueo alternativo por si, por cualquier circunstancia, no se pudiese reconocer al usuario. Este método alternativo puede ser "Patrón", "PIN" o "Contraseña".
833. En el caso de Android 7.x, la opción equivalente sigue estando disponible a través de Smart Lock desde el menú "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Smart Lock - [Reconocimiento facial] - Mejorar el reconocimiento facial".
834. Tras configurar una cara de confianza, desde el menú "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Smart Lock - Reconocimiento facial", el dispositivo móvil intentará identificar la misma cara del usuario cada vez que se intente desbloquear el dispositivo, procediendo a su desbloqueo si la cara es reconocida con éxito a través de la cámara frontal del dispositivo móvil, e indicándolo sutilmente con un icono con un candado abierto situado en la parte inferior de la pantalla de desbloqueo.
835. Uno de los principales inconvenientes de este método es que alguien con rasgos parecidos al usuario podría desbloquear el teléfono, o incluso emplear una fotografía o imagen animada del propietario del terminal para desbloquearlo.
836. Smart Lock no almacena fotos del usuario, sino datos biométricos para reconocer su cara que solo se almacenan en el dispositivo móvil, sin permitir que otras aplicaciones puedan acceder a ellos ni realizar copias de seguridad de los mismos en los servidores de Google [Ref.- 518].
837. Si la cara de confianza está activada, durante el proceso de reconocimiento de la cara, el candado de bloqueo de la parte inferior se transformará en símbolo de cara en lugar de como un candado. Cuando haya conseguido reconocerla, mostrará un candado parpadeante y, al deslizar el dedo hacia arriba, activará la pantalla. Esto puede permitir que un atacante aprenda que esta función está disponible.



838. Para añadir una cara de confianza, hay que entrar en "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Smart Lock" (se solicitará el código de acceso al dispositivo) - Reconocimiento facial - Configurar". Si esta opción está activa, en la pantalla de bloqueo aparecerá brevemente un icono con un símbolo de cara durante el tiempo en el que el terminal esté tratando de reconocer el rostro configurado. Si se consigue reconocerlo, el icono cambiará a un candado abierto que parpadea; en caso contrario, se mostrará un candado cerrado parpadeante (ver imágenes superiores).



839. Cuando se esté en condiciones adecuadas para registrar la cara del usuario, según las indicaciones del menú de Smart Lock, pulsar en "Siguiente" para iniciar el proceso de registro. Una vez completado, se puede optar por el menú "Mejorar reconocimiento facial" para registrar otras vistas de la cara y su aspecto en diferentes situaciones.



840. A partir de Android 5.1, se mejoró el proceso de desbloqueo del dispositivo móvil mediante la cara del usuario (Trusted Face), con un nuevo sistema de reconocimiento de caras, menos sensible a condiciones externas de luz, variaciones en la cara (con/sin gafas, con barba o afeitado, etc.), o a la posición del dispositivo móvil.

841. Desde el punto de vista de seguridad, el desbloqueo facial está desaconsejado, ya que, además de los aspectos reseñados anteriormente, se

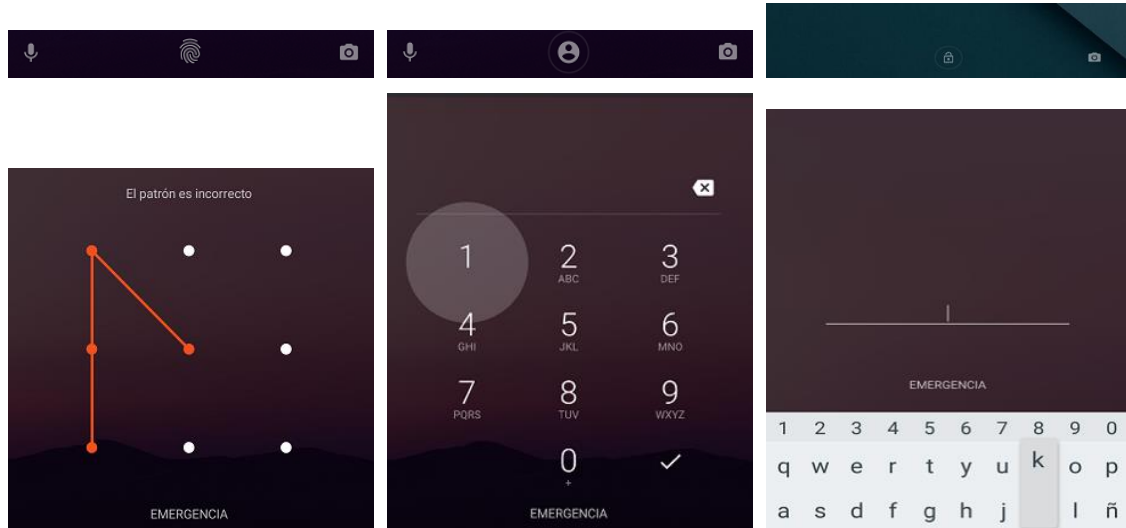
ha constatado que basta con que el usuario mire brevemente a la cámara (incluso involuntariamente) para que el dispositivo se desbloquee.

842. **"Reconocimiento de voz"**: este elemento de configuración permite desbloquear el dispositivo móvil en base al reconocimiento del patrón de voz del usuario mediante las capacidades asociadas al servicio "Ok Google". El servicio "Ok Google" permite iniciar búsquedas mediante la voz desde la aplicación de Google (o desde cualquier pantalla de inicio de Google Now Launcher), siendo necesario que este servicio esté configurado (ver apartado "Asistente de Google, Mi tablón y servicios de búsqueda de Google" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]).
843. En la integración de "OK Google" (o de desbloqueo por voz) con Smart Lock, y aunque al desbloquearse el dispositivo móvil mediante la voz se muestre en primer lugar la funcionalidad de "Ok Google", es posible seleccionar el botón para desplazarse a la pantalla previa, o el botón de acceso a la pantalla de inicio, para disponer de acceso completo al dispositivo móvil. Únicamente si se bloquea temporalmente, y de forma manual, el acceso mediante Smart Lock pulsando sobre el candado abierto disponible en la parte inferior de la pantalla de bloqueo, será necesario hacer uso del código de acceso en lugar de la voz para desbloquear de nuevo el dispositivo móvil.
844. En resumen, pese a la conveniencia que proporciona Smart Lock en la facilidad de uso del dispositivo móvil, esta funcionalidad no justifica los riesgos de seguridad asociados al acceso no autorizado al dispositivo por parte de un tercero en múltiples escenarios, por lo que se recomienda no activar Smart Lock y hacer únicamente uso de un código de acceso y, en todo caso, de autenticación mediante una huella dactilar digital, en caso de que la política de seguridad de la organización también contemple este método de acceso.
845. Adicionalmente, se han detectado vulnerabilidades en la propia funcionalidad de Smart Lock, por ejemplo, la descrita en el CVE-2016-6769 [Ref.- 207], que permitía a un potencial atacante con acceso físico al dispositivo móvil, acceder a la configuración de Smart Lock sin tener que introducir el código de acceso, concretamente en el caso en el que el usuario hubiera accedido al menú de Smart Lock como último ajuste de configuración.

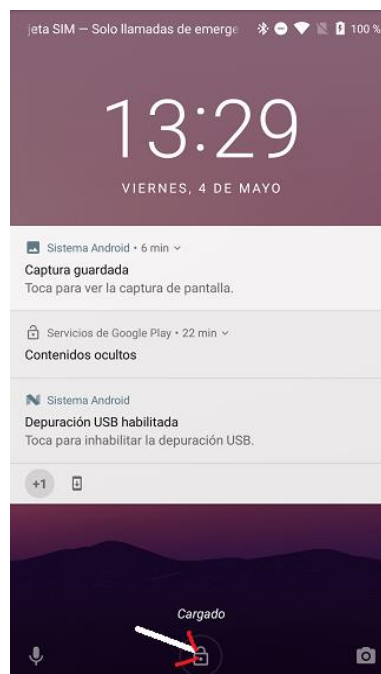
9.6. PANTALLA DE DESBLOQUEO: PREVISUALIZACIÓN DE DATOS

846. En primer lugar, como se puede apreciar en las siguientes imágenes, la pantalla de desbloqueo de Android permite diferenciar visualmente entre las diferentes configuraciones de desbloqueo del dispositivo móvil: ninguno - no se muestra ningún símbolo; deslizar - se muestra únicamente un candado; Smart Lock - se muestra un candado dentro de un círculo que parpadea; Smart Lock con desbloqueo facial - se muestra un icono de cara que parpadea; huella dactilar - se muestra una huella digital que parpadea; patrón, PIN, o contraseña - se muestra un candado dentro de un círculo

(además, en este caso, la pantalla de introducción del código de acceso también revelará cuál de los tres métodos está activo, pues se muestran los puntos de la matriz del patrón, un teclado numérico, o un teclado alfanumérico). Estos símbolos pueden facilitar a un potencial atacante una primera aproximación al nivel de seguridad existente en el dispositivo móvil:



847. La pantalla de desbloqueo del dispositivo móvil (*lock screen*) en Android permite, sin necesidad de introducir el código de acceso y, por tanto, sin necesidad de desbloquear el terminal, la previsualización del nombre del operador de telecomunicaciones en la parte izquierda de la barra superior de estado. En la zona central también se muestra, por defecto, la hora y la fecha, y el panel de notificaciones.



848. Adicionalmente, en la parte derecha de la barra superior de estado, se incluyen por defecto los iconos de conectividad (Wi-Fi y 2/3/4G y bluetooth), el nivel de batería y el usuario actual, además de iconos como el de "no molestar" si el dispositivo está en este modo.

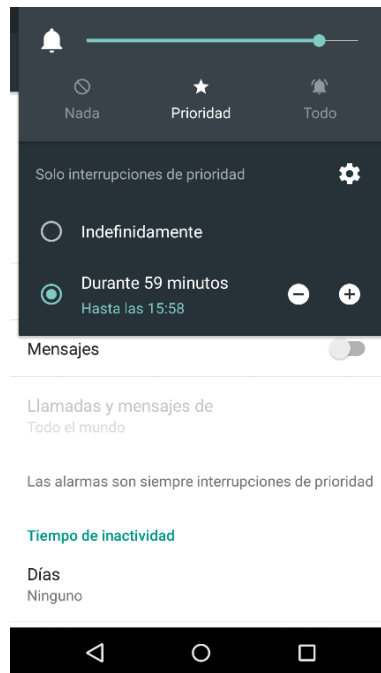
849. Con la introducción del configurador de IU de sistema (ver apartado "9.10. Configurador de IU del sistema"), se permite eliminar todos esos iconos de la barra de estado. Desde el punto de vista de seguridad, se recomienda prescindir de ellos para no desvelar información a un potencial atacante, que, con el simple acceso visual, podría saber si el dispositivo está conectado a una red Wi-Fi, si tiene habilitado el interfaz bluetooth o la ubicación, etc.
850. La pantalla de desbloqueo (a partir de Android 4.x) minimizó la posibilidad de desvelar información potencialmente confidencial y sensible a un potencial atacante. Por ejemplo, en el caso de recibirse un mensaje SMS, se indicaba mediante un aviso sonoro, una vibración y un icono de mensaje (por defecto), pero no se mostraba el contenido del mensaje ni en la pantalla de desbloqueo ni en la barra superior de estado.
851. Sin embargo, con el sistema de notificaciones (ver apartado "9.8. Notificaciones"), el usuario puede establecer que la pantalla de bloqueo muestre distintos grados de la información asociada a las aplicaciones (mostrar todo, ocultar todo, o mostrar sólo la información no sensible). Por ejemplo, con la opción "mostrar contenido de notificaciones", en la pantalla de bloqueo un SMS recibido se mostraría completo, incluyendo el número de teléfono del remitente. Con la opción "ocultar notificaciones sensibles", ante la recepción de un SMS sólo se mostraría el nombre de la app que gestiona los SMS y el texto "Contenidos ocultos":



852. Es por ello muy importante revisar la política de notificaciones para evitar que la pantalla de bloqueo pueda proporcionar información sensible que el usuario desearía ocultar.
853. Adicionalmente, la información desvelada en la pantalla de bloqueo puede extenderse mediante la utilización de *widgets* (ver apartado "9.13. Pantalla de desbloqueo: Widgets").
854. Asimismo, sigue siendo posible hacer fotografías y grabar vídeos sin desbloquear el dispositivo móvil (y sin conocer el código de acceso) deslizando hacia la izquierda el icono correspondiente a la cámara de fotos, situado en la esquina inferior derecha de la pantalla de bloqueo.
855. Con la incorporación del servicio "Asistente de Google" y, en particular, de "Ok Google" (éste último recientemente ampliado mediante el servicio "Voice Match") (ver apartado "Asistente de Google, Mi tablón y servicios de búsqueda de Google" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]), la pantalla de bloqueo añade un micrófono en la parte inferior izquierda que permite invocar comandos de voz que serán enviados al asistente de Google. Dado que estos comandos pueden realizar operaciones sensibles, se recomienda aplicar las recomendaciones mostradas en el correspondiente apartado para evitar comprometer la seguridad y la privacidad del usuario.

9.7. INTERRUPCIONES

856. Una interrupción es un evento que una aplicación decide enviar para informar al usuario de un hecho o situación relevante desde el punto de vista de la aplicación. El carácter de la interrupción es definido por la propia aplicación.
857. Una de las funciones que se incorporaron a partir de Android 5.x (Lollipop) son las interrupciones y notificaciones prioritarias, que permiten a los usuarios configurar el comportamiento del dispositivo móvil en función del calendario y de la importancia de la notificación para evitar interrupciones externas no deseadas.
858. En versiones previas de Android (4.x y anteriores), el modo "silencio" implicaba que el dispositivo móvil no emitía ningún sonido, ni vibraciones, pero sí admitía que las notificaciones visuales se mostrasen normalmente. Con la nueva funcionalidad añadida en Android 5.0, el modo silencio ("*None*" o "*Ninguno*"), no sólo silenciaba los sonidos y las vibraciones, sino que impedía por completo la llegada de alarmas y notificaciones, lo que limitaba mucho la usabilidad del dispositivo.
859. Android 5.1 resolvió el problema incluyendo dos modos de funcionamiento (además del modo "Todo", que no impone ninguna restricción): "Prioridad" y "Nada", así como su tiempo de duración. Para configurar el dispositivo en uno de estos modos manualmente, hay que pulsar el botón de volumen del dispositivo y se presentarán tres opciones:
- *Todo*: hasta nueva orden, o en el momento en que no cumplan las condiciones definidas, como por ejemplo el "Tiempo de inactividad", se permitirán todas las alarmas y notificaciones.
 - *Nada*: en este modo se suprimirán todas las interrupciones. Se permite seleccionar "Indefinidamente" (y sólo será posible salir del modo silencio manualmente) o "Durante X horas" (por defecto 1 hora, pero ampliable a las horas que especifique el usuario con los botones de + y -). Cuando se está en este modo, en la barra de estado (parte superior derecha de la pantalla) aparecerá un símbolo de prohibición.
 - *Prioridad*: al igual que en el modo anterior, se puede especificar un tiempo indefinido o una duración en horas. Además, en la parte superior del menú, aparece el menú de ajustes de las interrupciones de prioridad, que equivale a entrar en el menú "Ajustes - [Dispositivo] Sonido y notificaciones - [Sonido] Interrupciones". Cuando el dispositivo se encuentra en este modo, en la barra de estado (parte superior derecha de la pantalla) aparecerá una estrella. El propósito de este modo es filtrar las notificaciones menos importantes en los períodos de tiempo definidos. Las alarmas siempre se consideran interrupciones prioritarias (o de prioridad).



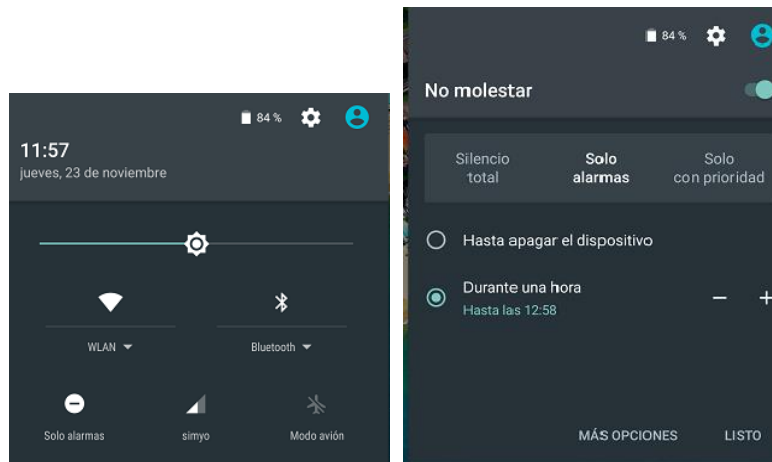
860. Android 5.1 introdujo el concepto de "interrupciones de prioridad", que son aquellas que pueden disparar acciones en un dispositivo móvil que ha entrado en el modo "prioridad"; a nivel de aplicación, una interrupción de prioridad puede enviar notificaciones; a nivel de contacto, un contacto de prioridad podrá enviar mensajes o cursar llamadas. El resto de notificaciones y llamadas quedarán silenciadas.

861. Android 6.x complementó el modelo de prioridades que surgió con Android 5.1 introduciendo el concepto de modo "No molestar" e independizando las alarmas de las demás interrupciones consideradas "de prioridad". Si Android 5.1 incluyó los modos "Prioridad" y "Nada" (además del modo "Todo", que no impone ninguna restricción), Android 6.x introdujo los modos "*Silencio total*" (equivalente a "*Nada*" en Android 5.1), "*Solo alarmas*" (para que únicamente las alarmas interrumpieran al usuario) y "*Solo con prioridad*" (equivalente al modo "*Prioridad*" de Android 5.1).

862. Para configurar el dispositivo móvil en modo "No molestar", es necesario deslizar el dedo desde la parte superior de la pantalla hasta que aparezca el panel de ajustes de configuración rápidos. Seleccionando el icono "No molestar", se activa esa opción y se presenta un menú que da a elegir entre los diferentes modos disponibles. Por defecto, la opción activa es "Solo alarmas":

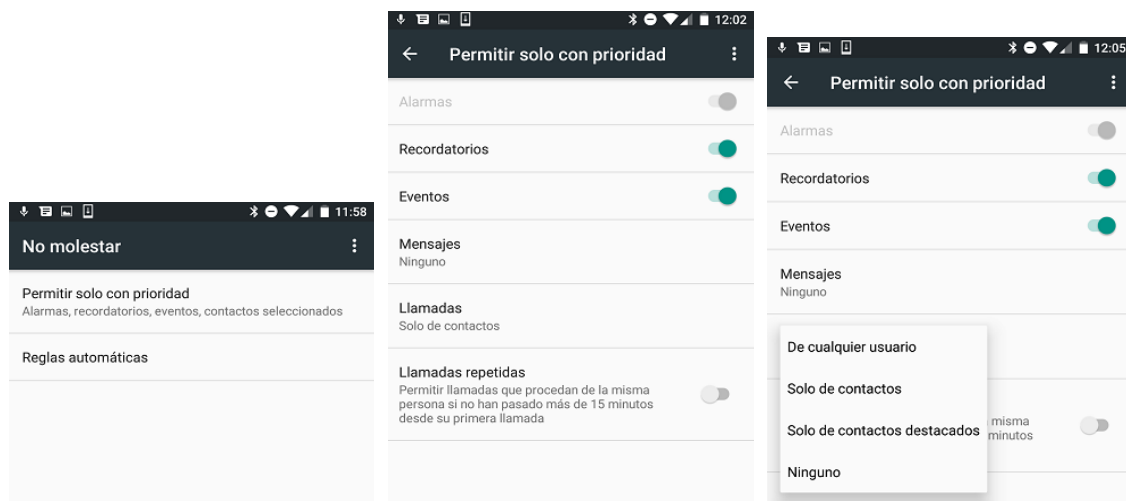
- "Silencio total": en este modo se suprimirán todas las interrupciones, bloqueándose cualquier sonido o vibración del dispositivo móvil. Se permite seleccionar "Hasta apagar el dispositivo" (y sólo será posible salir del modo silencio manualmente) o "Durante X horas" (por defecto 1 hora, pero ampliable a las horas que especifique el usuario con los botones de + y -). Cuando se está en este modo, en la barra de estado (parte superior derecha de la pantalla) aparecerá un símbolo de prohibición.
- "Solo alarmas": se dispone de las mismas opciones que en el modo "silencio total", pero se permiten las notificaciones o interrupciones de las alarmas.

- "Solo con prioridad": la primera vez que se selecciona este modo, el dispositivo mostrará un mensaje informativo y ofrecerá la opción "Personalizar". En veces sucesivas, se mostrarán las mismas opciones que para el resto de los modos.



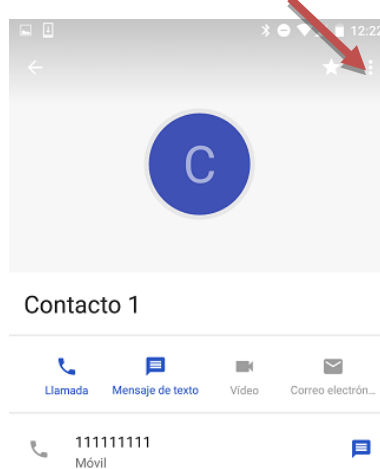
863.Si, en cualquiera de los modos se pulsa sobre "Más Opciones", se abrirá el menú "Ajustes - [Dispositivo] Sonido y notificaciones - [Sonido] No molestar" (descrito posteriormente). Se solicitará el código de acceso al dispositivo si el mismo se encontraba en la pantalla de bloqueo.

864.En Android 7.x las acciones para configurar las interrupciones han cambiado de formato respecto a Android 5.1. Es posible configurar las reglas de las interrupciones desde el menú "Ajustes - [Dispositivo] Sonido y notificaciones - [Sonido] No molestar". Al seleccionar el menú aparecen las siguientes opciones:

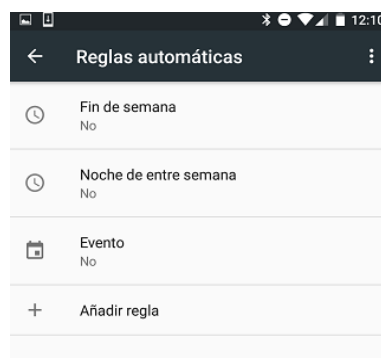


- "*Permitir solo con prioridad*": seleccionando este parámetro se desplegará un menú con las condiciones a cumplir por parte de una interrupción para que se la considere de prioridad (las alarmas, aunque se listan como opción, siempre son consideradas de prioridad y su botón de activación está sombreado):
 - Recordatorios.
 - Eventos.

- Mensajes: ninguno, solo de contactos, solo de contactos destacados, de cualquier usuario. Para destacar un contacto, es necesario marcar el símbolo "estrella" en los detalles del contacto:

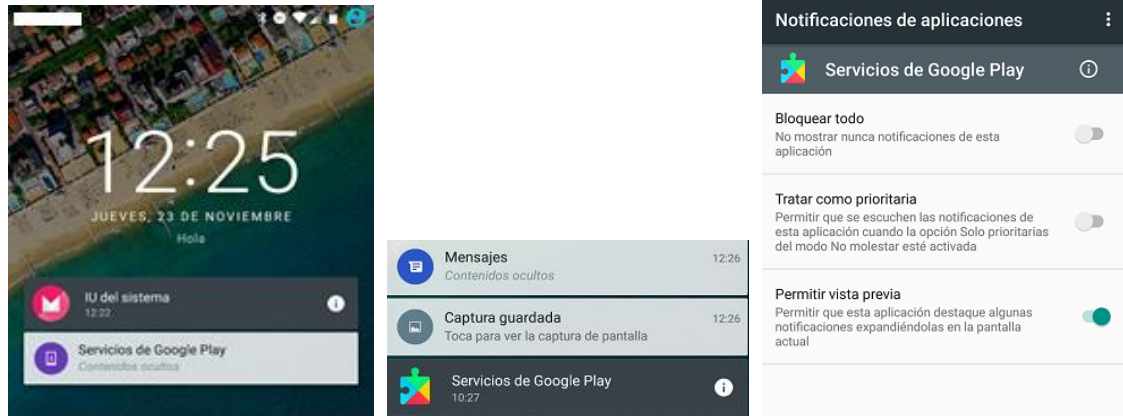


- Llamadas: las opciones son iguales que para "mensajes".
- Llamadas repetidas: esta opción es nueva en la versión 6.x de Android, y permite que se cursen llamadas que se repiten en un margen inferior a 15 minutos, aunque no cumplan la condición establecida en el campo "Llamadas".
- "*Reglas automáticas*": permite definir reglas temporales por las que el dispositivo entrará en el modo de interrupción definido por la regla. Se presentan tres reglas por defecto (inactivas): "fin de semana", "noche de entre semana" y "evento". Además, se muestra la opción "Añadir regla", que permite al usuario crear sus propias reglas de tipo "evento" o "de hora". Las reglas de hora, incluidas las de "fin de semana" y "noche de entre semana", presentan las siguientes opciones:
 - Nombre de la regla.
 - Días en los que la regla se activará.
 - Hora de inicio de la regla.
 - Hora de finalización de la regla.
 - No molestar: a elegir entre "solo prioritarias", "solo alarmas" o "silencio total".

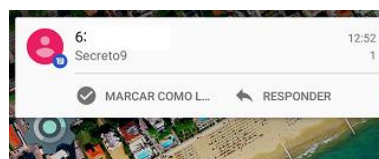


9.8. NOTIFICACIONES

865. Una notificación es un mensaje que una aplicación puede mostrar al usuario fuera del interfaz gráfico propio de la aplicación móvil, para informarle de un evento que se considera relevante desde el punto de vista de la aplicación.
866. Las notificaciones se envían al sistema para que las procese de acuerdo a la configuración definida por el usuario en relación a la política de interrupciones que se encuentre activa en el dispositivo móvil en el momento de recibirse la notificación, a la aplicación que envía la notificación, y a la prioridad de la propia notificación.
867. Android 5.0 (Lollipop) introdujo un cambio en el paradigma de las notificaciones [Ref.- 508], entre ellas la capacidad para que una notificación aparezca en la pantalla de bloqueo del dispositivo o que aparezca como una pequeña ventana flotante (*heads-up notification*) cuando el dispositivo móvil está desbloqueado. Además, las notificaciones flotantes pueden mostrar botones de acción que permiten al usuario manejar la notificación sin salir de la aplicación que está activa, o simplemente descartar la notificación.
868. Las notificaciones flotantes surgen cuando la aplicación activa está en modo de pantalla completa y cuando la notificación tiene carácter prioritario y utiliza tonos o vibraciones.
869. Sin embargo, con Android 5.0 surgía el problema de que, cuando llegaba una notificación, no sólo se mostraba en primer plano independientemente de lo que el usuario estuviese haciendo, sino que, además, las únicas opciones para actuar ante ella eran: esperar a que desapareciera transcurridos 10 segundos, procesarla inmediatamente, o descartarla deslizando hacia los lados (derecha o izquierda) de la pantalla.
870. En Android 5.0, el comportamiento de las notificaciones es lo que se denomina "emergente", es decir, la notificación se presenta en la pantalla activa interrumpiendo al usuario.
871. Con la llegada de Android 5.1, y todavía vigente en Android 6.x, las *heads-up notifications* se empezaron a mostrar, con el dispositivo móvil desbloqueado, en la barra de notificaciones en lugar de en mitad de la pantalla. Con el dispositivo bloqueado, se muestran en mitad de la pantalla. Al igual que en versiones previas, se permite descartar la notificación (deslizando hacia cualquiera de los lados), pero, adicionalmente, se puede minimizar deslizando el dedo desde la notificación hacia arriba, lo que la desplaza al panel o barra de notificaciones (situado en la parte superior) sin perder su contenido, hasta que el usuario decida procesarla.
872. Realizando una pulsación prolongada sobre una notificación, tanto con la pantalla bloqueada como desbloqueada, se mostrará qué componente o app ha generado la notificación. El icono de (i) (información), al pulsarse, abrirá el menú de configuración de las notificaciones de la aplicación concreta, permitiendo al usuario elegir cómo tratar las notificaciones de esa aplicación:



873. Con la pantalla desbloqueada, la app puede ofrecer adicionalmente acciones como respuesta a la notificación:



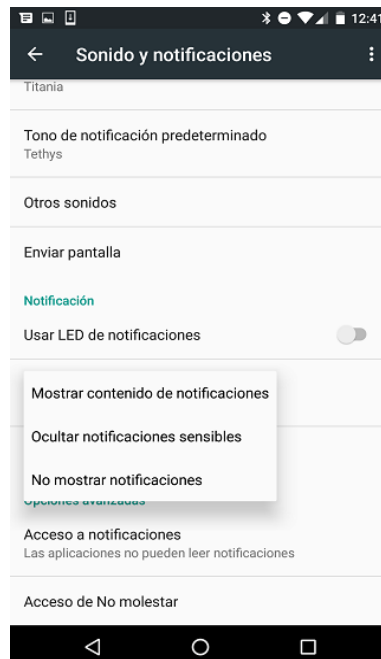
874. El servicio de notificaciones está afectado directamente por la configuración de las interrupciones (ver apartado "9.7. Interrupciones").

875. La presentación de notificaciones se puede configurar a varios niveles:

- Tratamiento de la notificación: se ve afectado por el estado del dispositivo móvil (bloqueado frente a desbloqueado) y por el nivel de interrupción que esté activo (ver apartado "9.7. Interrupciones").
- Sensibilidad de la notificación (se define a nivel de la aplicación móvil).
- Definición de la prioridad de la notificación de cada aplicación (carácter de la notificación).

876. El tratamiento de una notificación "Con dispositivo bloqueado" se configura a través del menú "Ajustes [Dispositivo] - Sonido y notificaciones - [Notificación] Con dispositivo bloqueado". Por un lado, muestra el modo en que se gestionan las notificaciones según la configuración actual y, al pulsar sobre ella, permite definir cómo se gestionarán las notificaciones en la pantalla de desbloqueo, desplegando un menú cuyas opciones son:

- "*Mostrar contenido de notificaciones*": visibilidad pública del contenido de la notificación.
- "*Ocultar notificaciones sensibles*": sólo se mostrará parte de los contenidos; visibilidad privada.
- "*No mostrar notificaciones*": si no se desea que se muestren, ni siquiera el icono o mensaje de la notificación; visibilidad secreta.



877. Con el dispositivo bloqueado, las notificaciones que cumplan la condición para ser visualizadas según el modo de interrupción activo actualmente se mostrarán en el área de notificaciones, que se muestra en primer plano en la pantalla de desbloqueo. Se mostrarán un total de cuatro notificaciones, y, en caso de haberse recibido más, el resto se englobarán en una sola entrada que va precedida con el símbolo "+número", siendo "número" el número de notificaciones activas que no se están visualizando actualmente.



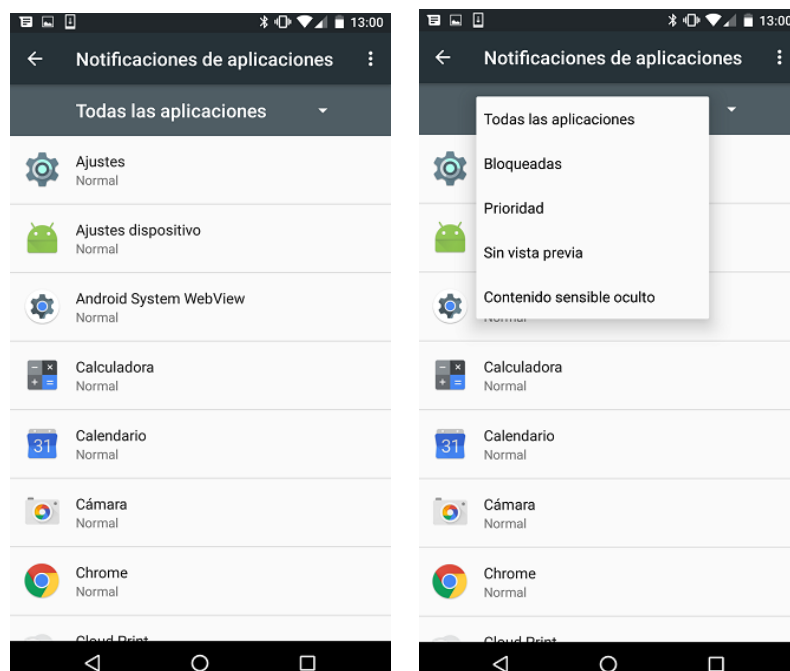
878. Con el dispositivo desbloqueado, las notificaciones se mostrarán en el panel de notificaciones (*notification drawer*) situado en el margen superior izquierdo del dispositivo móvil y, para ver el detalle de la notificación, el usuario ha de desplegar el panel deslizando el dedo desde la parte superior hacia abajo. Tanto el área de notificaciones (de la pantalla de desbloqueo), como el panel de notificaciones, son áreas controladas por el sistema que el

usuario puede ver en todo momento, por lo que su contenido debe controlarse para evitar problemas de seguridad y privacidad (por ejemplo, algunos servicios, como pasarelas de pago, envían al usuario un SMS con el código que han de introducir en la aplicación para validar el pago o completar la operación (segundo factor de autenticación). Un potencial atacante podría explotar una vulnerabilidad que permita el secuestro de la sesión del usuario en dicho servicio y, teniendo acceso visual a su dispositivo móvil, podría obtener y validar el código recibido y, por tanto, realizar la operación deseada.

879. Android 6.x (Marshmallow) refinó el tratamiento de las notificaciones en relación a su antecesor, Android 5.1 (Lollipop) permitiendo mayor control sobre las notificaciones tanto a nivel de cómo se muestran en pantalla al ser recibidas, como a nivel de su configuración por aplicación.

880. De manera global, el comportamiento del sistema ante una notificación de una aplicación se define en el menú "Ajustes [Dispositivo] - Sonido y notificaciones - [Notificación] Notificaciones de aplicaciones".

881. Al pulsar sobre "Notificaciones de aplicaciones", se despliega un menú que ordena las aplicaciones alfabéticamente por nombre de aplicación. Por defecto, se muestran todas las aplicaciones, pero es posible filtrar en base al comportamiento definido por el usuario para cada una de ellas:

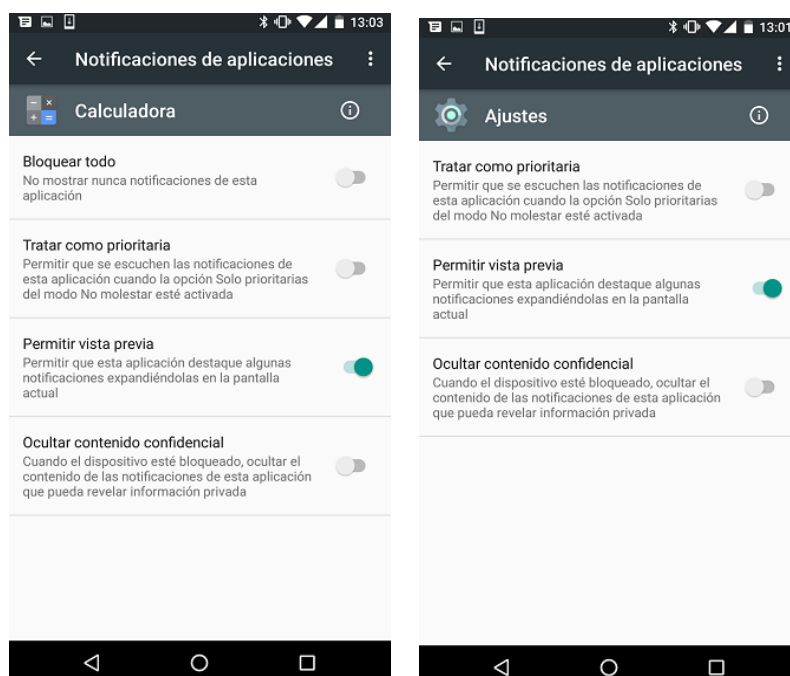


882. Para definir la acción que se tomará con respecto a las notificaciones de una aplicación concreta, se debe pulsar sobre el nombre de la aplicación, y aparecerá un menú con las siguientes opciones:

- Icono de información (i) junto al nombre de la aplicación: abre los ajustes de la aplicación de igual modo que el menú "Ajustes - [Dispositivo] Aplicaciones - <nombre de la aplicación>". Uno de los ajustes del citado menú es "Notificaciones", que, al ser seleccionado, mostrará las opciones

de tratamiento de las notificaciones de esta aplicación por parte del sistema, enumeradas justo a continuación.

- Bloquear todo: no se mostrarán notificaciones de esta aplicación. Esta opción no está disponible para ciertas aplicaciones de sistema como, por ejemplo, la aplicación "Ajustes".
- Tratar como prioritaria: cuando el dispositivo móvil esté configurado para aceptar sólo interrupciones prioritarias, únicamente las notificaciones que tengan este carácter se aceptarán cuando el modo "No molestar" lo permita (ver apartado "9.7. Interrupciones").
- Permitir vista previa: la aplicación así catalogada está autorizada para destacar y expandir notificaciones en la pantalla del dispositivo móvil.
- Ocultar contenido confidencial: con el dispositivo móvil bloqueado, se ocultará el contenido que la aplicación marque como privado o sensible.



883. Es importante destacar que en Android 7.x la sensibilidad de las notificaciones (y, por tanto, su presentación en la pantalla de bloqueo) la define la app que genera la notificación a través del método "setVisibility()" del intent asociado a la notificación [Ref.- 628]. Los posibles valores son *public* (el contenido de la notificación se mostrará en su totalidad incluso en la pantalla de bloqueo), *secret* (el contenido se ocultará en la pantalla de bloqueo, aunque el desarrollador puede elegir reemplazar el contenido por otra información no sensible) y *private* (la notificación mostrará información básica, como un título o un icono, pero no su contenido completo).

884. Al hilo de lo anterior, Android 7.x no proporciona ninguna opción que permita al usuario alterar la sensibilidad de las notificaciones de una aplicación concreta. Por ejemplo, no se puede especificar que se desea que las notificaciones de la app que gestiona los SMS sean siempre tratadas como privadas. Únicamente se puede elegir si se oculta el contenido de las

notificaciones que la app marque como privadas o secretas, o si se bloquearán por completo.

885. Android 4.4 (KitKat) introdujo un permiso denominado "Acceso a notificaciones", que una app puede solicitar al usuario al lanzarse su ejecución. El usuario deberá otorgar el permiso de forma explícita, y ninguna app lo tiene concedido de forma automática. Es importante vigilar qué aplicaciones lo solicitan y a cuáles se les concede, ya que activarlo pone en riesgo la privacidad del usuario. Puesto que las aplicaciones con acceso a las notificaciones pueden acceder a todas las notificaciones sin distinción, podrían también acceder a notificaciones especialmente sensibles, como un mensaje confidencial o crítico. Podría darse el caso de que una aplicación maliciosa capturase las notificaciones y las utilizase con propósitos maliciosos, dañinos o deshonestos, por ejemplo, capturar un SMS enviado por una app de banca on-line cuyo contenido incluya el código de autorización de una transacción.
886. Para identificar qué apps han obtenido el permiso de "Acceso a notificaciones", se debe entrar en esa opción a través de "Ajustes - [Dispositivo] Sonido y notificaciones - [Opciones avanzadas]".
887. Google recomienda a los desarrolladores de apps manejar las notificaciones de forma inteligente: por ejemplo, en lugar de reiterar una notificación para un mismo tipo de evento, se aconseja actualizar las notificaciones previas (*stacking notifications*). Un escenario típico de *stacking* sería, para aplicaciones de correo, si llega un nuevo mensaje, cambiar la notificación anterior para que incremente el número de mensajes pendientes sin leer, en lugar de mostrar una nueva notificación adicional indicando que ha llegado otro correo [Ref.- 507].

9.9. AHORRO DE BATERÍA (DOZE)

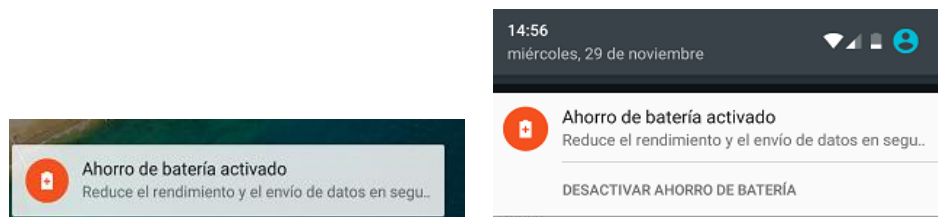
888. Android 6.x (API 23) introdujo el denominado proyecto "Doze", cuyo propósito es establecer diferentes comportamientos para las apps en función de si el dispositivo móvil está conectado a la corriente, usando únicamente la batería, o cuando el mismo no está siendo utilizado por el usuario, especialmente durante largos periodos de tiempo [Ref.- 616], a fin de optimizar el consumo de batería.
889. Si el usuario deja el dispositivo móvil sin alimentación de corriente, con la pantalla apagada, y sin moverlo (inmóvil), durante un periodo de tiempo determinado, éste entrará en un estado de ahorro de energía conocido como modo Doze.
890. En este estado el dispositivo permanece dormido o suspendido, y resume periódicamente su ejecución durante breves periodos de tiempo para que el sistema pueda completar operaciones pendientes y que las apps puedan sincronizar datos.
891. En el modo Doze se establecen dos niveles de ahorro cuando se apaga la pantalla de un dispositivo no conectado a la corriente, cada uno de los cuales impone una serie de restricciones:

- Modo suspendido (*App Standby*): en este modo, la actividad de red y las actividades de sincronización para las apps que no han sido utilizadas recientemente por el usuario se pospone, estableciéndose unas ventanas de mantenimiento en las que se procesarán las actividades pendientes. Se considera que una aplicación no se ha utilizado recientemente cuando el usuario no ha pulsado sobre la pantalla de la app y no se cumple ninguna de las siguientes condiciones:
 - La app no tiene ningún proceso en primer plano en el momento actual (por ejemplo, una app que permite escuchar música de fondo aunque el reproductor multimedia no esté en primer plano).
 - La app ha lanzado una notificación que está activa.
 - El dispositivo móvil está conectado a la corriente.
 - Modo descanso (*Doze*): cuando el dispositivo permanece inmóvil (*standby*) y con la pantalla apagada durante un periodo de tiempo determinado, el sistema operativo restringirá el acceso de las apps a la red (incluyendo los escaneos GPS y Wi-Fi) y a otros recursos del dispositivo móvil (concretamente, los eventos *PowerManager.WakeLock*, utilizados para indicar que la app requiere que el dispositivo se mantenga activo, y *AlarmManager*, que proporciona acceso a los servicios del sistema de alarmas).
892. Android 7.x incorpora nuevas mejoras a Doze, denominadas "Doze on the Go", aplicando una serie de restricciones de acceso a la CPU y a la red mientras el dispositivo está desconectado de la corriente y con la pantalla apagada, pero sin requerir que esté completamente inmóvil (por ejemplo, puede estar en el bolsillo del usuario y éste en movimiento).
893. Android 7.x, siguiendo las directrices del proyecto Svelte introducido en Android 4.4 para mejorar el rendimiento de los dispositivos móviles, contribuye al mejor uso de la batería suprimiendo tres tipos de broadcast implícitos, los cuales suelen despertar a las apps que registraron su interés en ellos cuando ejecutan en segundo plano:
- **CONNECTIVITY_ACTION**: este broadcast se dispara cuando hay cambios en el tipo de conexión (por ejemplo, cambio de red Wi-Fi a red de datos móviles). Esta restricción afecta a las apps con API nivel 24, que sólo escuchan eventos de cambios de red a través de su hilo principal, activo cuando se usa la app.
 - **ACTION_NEW_PICTURE** y **ACTION_NEW_VIDEO**: este broadcast se suprime para todas las apps, incluyendo las anteriores a la API 24.
894. Una aplicación de tipo administrador de dispositivos nunca entra en modo suspendido.
895. Para activar el modo de ahorro de batería, el usuario debe acceder al menú "Ajustes - [Dispositivo] Batería ... - Ahorro de batería". En la versión de Android 7.1.2 el interruptor sólo puede desplazarse a la posición "Sí" cuando el dispositivo no se encuentra conectado a la corriente. La selección del

ajuste "Activar automáticamente" permite establecer los umbrales de porcentaje de carga de la batería:



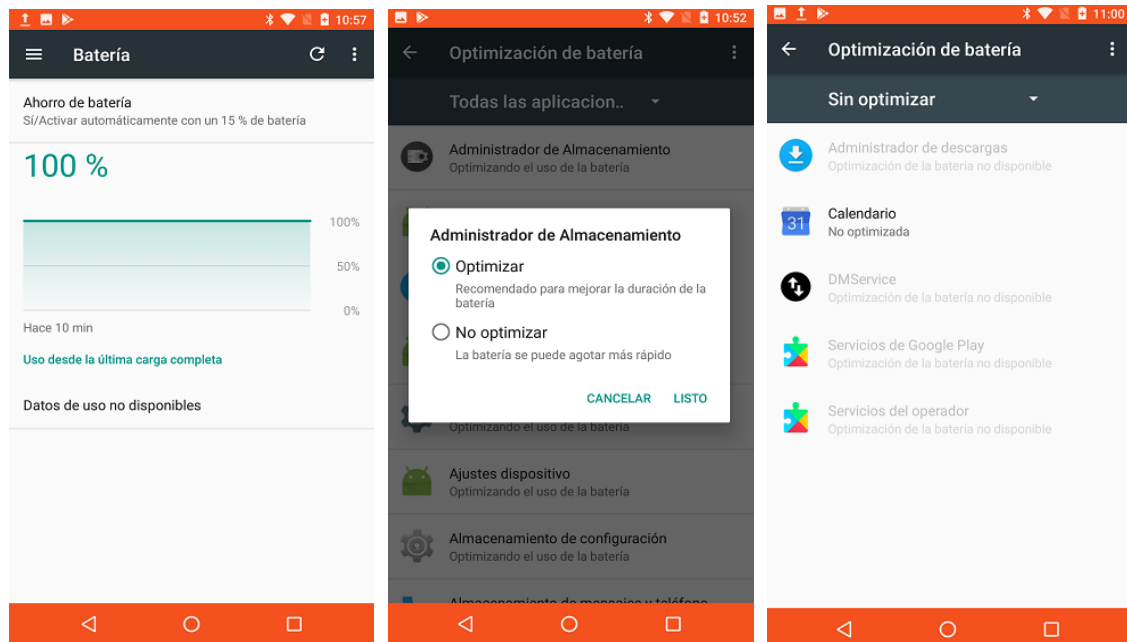
896. En Android 6.x, al activarse este modo de ahorro por parte del usuario, se muestra una notificación que permanecerá activa en la barra de notificaciones con un símbolo de batería y el signo "+" (este comportamiento se ilustra en las imágenes siguientes). Esta notificación, que no se puede descartar, admite como respuesta la opción "Desactivar ahorro de batería" y, al pulsarla, proporciona acceso al menú de ahorro de batería. Sin embargo, en Android 7.x no se muestra dicha notificación.



897. El dispositivo, con la pantalla activa, presentará unas bandas rojas en la barra de notificaciones y en los botones táctiles de la pantalla.

898. Se establece una lista de exclusión (o *whitelist*) que permite al usuario definir qué apps pueden saltarse parcialmente las políticas de los modos Doze y App Standby. Para ello, el usuario puede acceder a "Ajustes - [Dispositivo] Batería - [...] - Optimización de batería". Por defecto, se mostrará la pantalla de "Sin optimizar", que engloba las apps que están en la *whitelist*. En gris aparecen las apps que nunca pueden entrar en modo ahorro.

899. Si se desea añadir una app a esta lista, se desplegará el menú "Sin optimizar" y se elegirá la opción "Todas las aplicaciones". Con ello se mostrarán todas las apps instaladas. Una vez seleccionada la app a incluir en la lista de exclusión, se marcará "No optimizar".

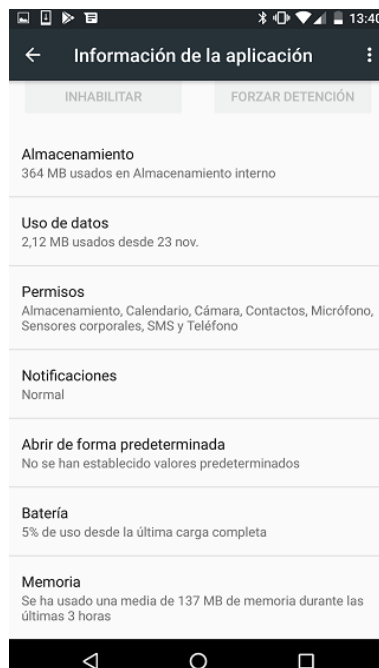


900. Una app no puede saltarse el mecanismo de ahorro de batería si éste está activo. No obstante, Android establece una serie de usos aceptables ante los cuales la app podrá:

- Lanzar un `ACTION_IGNORE_BATTERY_OPTIMIZATION_SETTINGS`: este intent abrirá para el usuario el menú de optimización de batería descrito previamente, posibilitando que se pueda añadir dicha app.
- Solicitar el permiso `REQUEST_IGNORE_BATTERY_OPTIMIZATIONS` para permitir al usuario añadir la app a la lista de exclusión directamente, sin pasar por el intent anterior.

901. Ninguno de los dos elementos anteriores provocará que la app sea añadida por el sistema a la lista de exclusión sin intervención del usuario. Además, los usos aceptables de ambos mecanismos están descritos oficialmente en la sección "Acceptable Use Cases for Whitelisting" del documento "Optimizing for Doze and App Standby" [Ref.- 616].

902. Es posible saber cuál es el consumo de batería de una aplicación a través del campo "Batería" de la sección "Ajustes - [Dispositivo] Aplicaciones - [Nombre de la app] Información de la aplicación". Esta información es muy útil de cara a identificar problemas de consumo irregular de batería.



903. Desde el punto de vista de la seguridad del dispositivo móvil, se recomienda mantener un nivel de batería aceptable que no impida, en caso de pérdida, localizar el dispositivo, bien a través de los servicios de localización remota (ver apartado "8.1. Gestión remota de dispositivos Android por parte del usuario"), bien por medios más básicos como llamar al número de teléfono. En este sentido, el uso de las capacidades de ahorro de batería resulta beneficioso para el usuario y para el hardware del dispositivo móvil.

9.10. CONFIGURADOR DE IU DEL SISTEMA

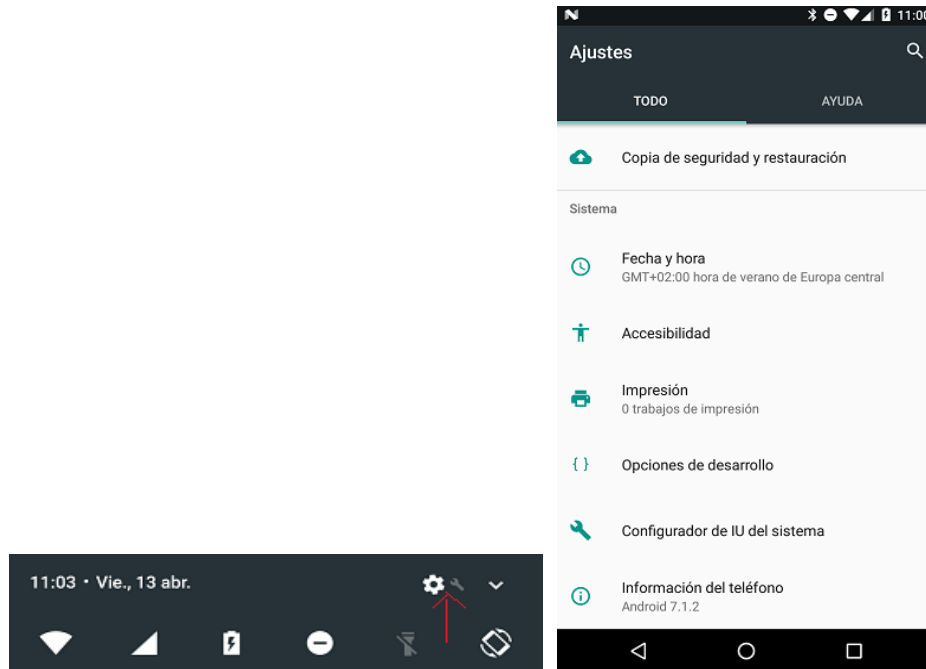
904. El "Configurador de IU del sistema" (*System UI Tuner*) es una característica introducida en Android 6.x (Marshmallow) que da acceso a funcionalidades "ocultas" que permiten personalizar ciertos elementos del interfaz de usuario de Android, en particular, la barra de estado y el menú de ajustes rápidos (disponible en la parte superior del interfaz gráfico).

905. Hay que reseñar que algunas de estas funcionalidades no están soportadas oficialmente por Google y pueden variar entre versiones. En caso de detectarse un empeoramiento en el rendimiento o en la estabilidad del sistema, se debe proceder a deshabilitarlas.

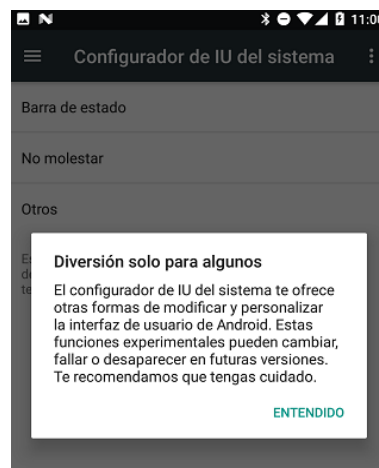
906. Para activar el configurador de IU (Interfaz de Usuario), hay que habilitar el modo "desarrollador" en el dispositivo móvil (consultar apartado "13.3. Opciones de desarrollo y depuración USB" para más información sobre este modo). Una vez habilitado, es necesario desplegar la barra de ajustes de configuración rápidos y realizar una pulsación prolongada sobre el icono del engranaje correspondiente a los ajustes (cuando este icono empiece a girar, se mantendrá la pulsación durante unos 10 segundos).

907. Con esta acción, se añadirá un nuevo menú bajo "Ajustes - [Sistema] <Símbolo de llave inglesa> Configurador de IU del sistema". Además, el

símbolo de engranaje que representa a los ajustes en la barra superior de estado se mostrará con un pequeño icono de llave inglesa a su derecha:

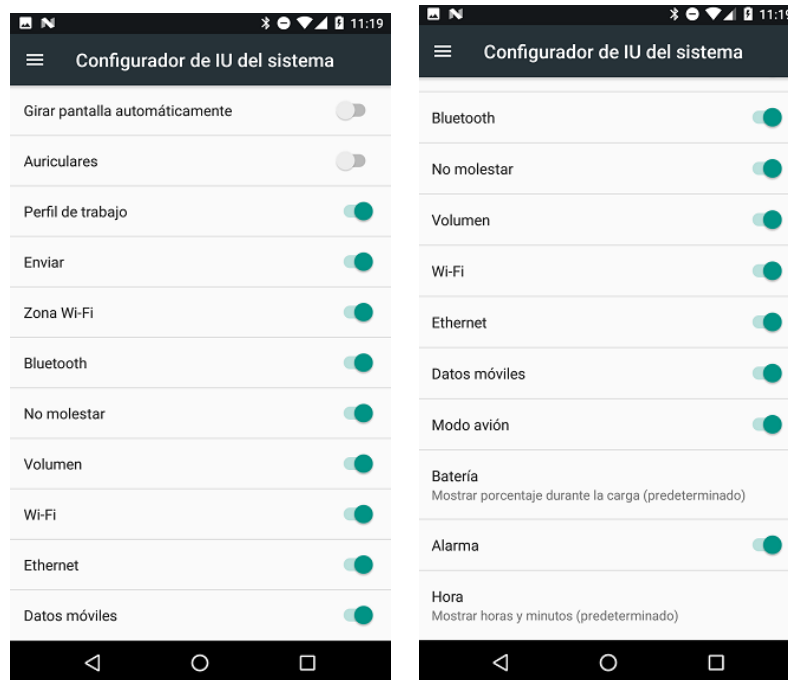


908.Una vez activo, el menú "Configurador de IU del sistema" permite personalizar el interfaz de usuario (IU). Pese a ello, no se recomienda hacer uso de su funcionalidad a menos que se esté seguro de lo que se persigue con ello, de lo cual se advierte al usuario a través del siguiente mensaje:

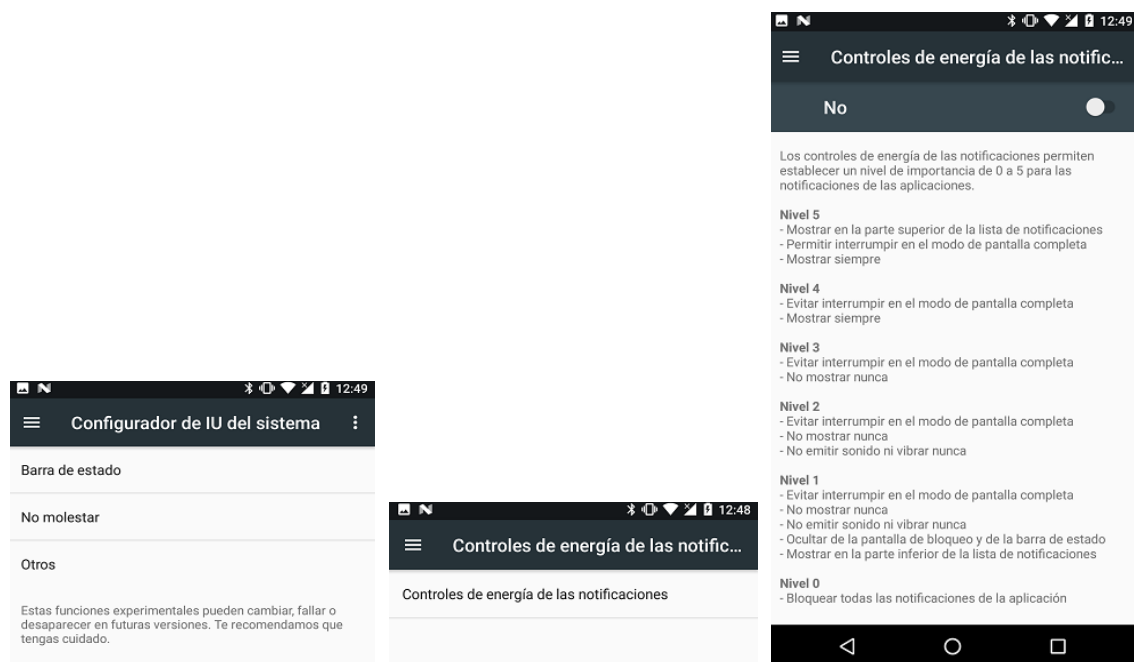


909.En Android 6.x, la personalización del menú de ajustes rápidos descrita en el apartado "9.12. Menú de ajustes rápidos y barra de notificaciones" se llevaba a cabo desde el configurador IU de sistema. Sin embargo, en Android 7.x este menú se configura según se ilustra en dicho apartado.

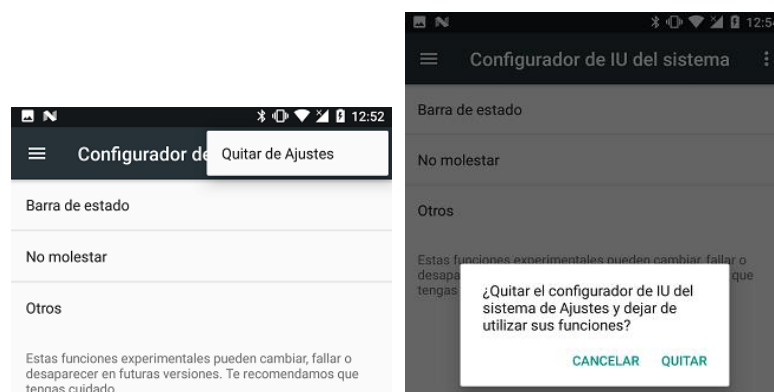
910.La opción "Barra de estado" permite configurar los iconos correspondientes a las opciones que se sitúan en la esquina superior derecha de la pantalla cuando se den las condiciones asociadas a ellos, y ofrece las siguientes opciones:



911. Es importante reseñar que la barra de estado no mostrará un icono si la funcionalidad asociada a él no aplica en un momento concreto. Así, si los auriculares no están conectados, aunque el interruptor "Auriculares" esté activo, no se verá su icono.
912. La opción "Batería" hará que en el icono de batería de la barra de estado aparezca un número representativo del porcentaje de carga actual, aún con el dispositivo móvil sin conectar a una fuente de carga.
913. Algunos ajustes (como "Batería" y "Hora") no se representan con un interruptor. Al pulsar sobre ellos, se desplegará un menú que permitirá definir el formato de presentación de esa funcionalidad en la barra de estado.
914. En Android 7.x desaparece la opción "Modo de demostración" que introdujo Android 6.x, y que reemplazaba temporalmente la barra de estado por una totalmente limpia de notificaciones, mostrando únicamente los iconos de Wi-Fi, cobertura y batería al 100%, junto a las 6:00 como la hora actual, en clara referencia a Android 6.0 (Marshmallow).
915. Además del menú "Barra de estado", en Android 7.x, el menú "Configurador de IU del sistema" ofrece la opción "Otros", que en Android 7.1.2 sólo alberga la opción "Controles de energía de las notificaciones" (consultar el apartado "9.8. Notificaciones" para más información):



916. Para deshabilitar el configurador de IU de sistema, se puede ir a "Ajustes - Sistema - Configurador de IU del sistema", seleccionar los 3 puntos de la derecha y marcar "Quitar de ajustes":

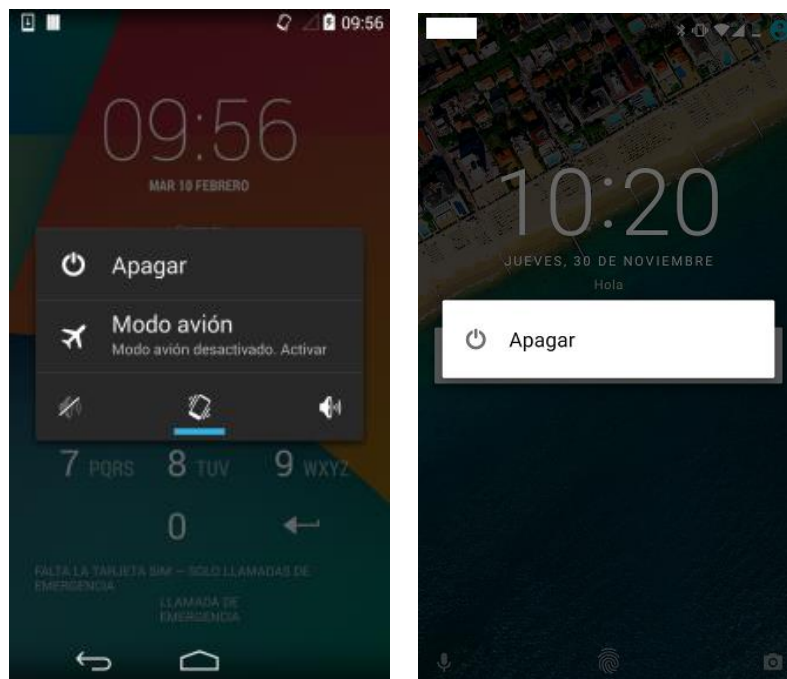


917. Aunque sería deseable poder activarlo para realizar las personalizaciones necesarias y desactivarlo después para evitar los potenciales riesgos de estabilidad que puede entrañar usarlo, resulta que su desactivación conlleva la pérdida de cualquier personalización realizada previamente.

9.11. PANTALLA DE DESBLOQUEO: APAGAR EL DISPOSITIVO MÓVIL

918. Desde la pantalla de desbloqueo de Android, y sin conocer el código de acceso pero disponiendo de acceso físico temporal al dispositivo móvil, es posible mantener pulsado el botón de encendido y apagado, para proceder a apagar el terminal.

919. Android 4.x ofrecía a través del botón de encendido, además de la opción "Apagar", un menú que permitía gestionar el sonido, indicando si el dispositivo móvil está en silencio, en modo vibración, o con el volumen activo. También incluía la opción de "Modo avión". Todos estos controles desaparecieron de este menú en Android 5.x, quedando únicamente la opción "Apagar" también para Android 6.x.



Menú de apagado en Android 4.x vs. Menú de apagado en Android 6.x/7.x

Nota: En Android 4.x, el botón de apagar ofrecía adicionalmente activar el modo avión (o modo vuelo, botón "Modo avión"), lo que suponía un riesgo desde el punto de vista de seguridad ya que, si una persona no autorizada obtiene acceso físico al dispositivo móvil y activa el modo avión, no será posible disponer de conectividad de datos desde el dispositivo móvil para poder hacer uso de las capacidades remotas de gestión del mismo (ver apartado "8.1. Gestión remota de dispositivos Android por parte del usuario"), por ejemplo, en escenarios donde se ha perdido el dispositivo móvil, o éste ha sido robado⁶⁸.

Android 5.x no resuelve este problema, porque, si bien la opción para activar/desactivar el modo avión desaparece del botón de encendido, se incorpora irreversiblemente al menú de ajustes de configuración rápidos, que son accesibles como interruptores sin necesidad de autenticación.

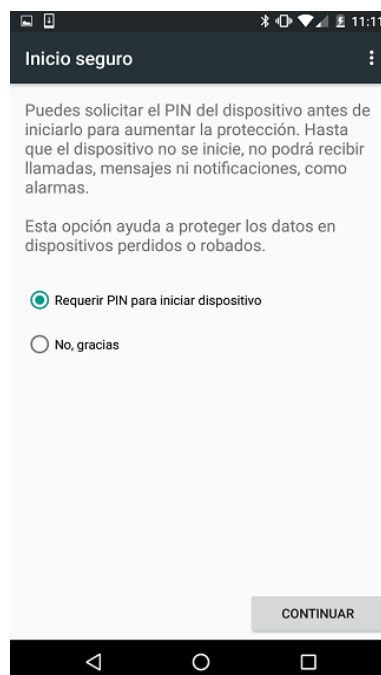
920. Android 6.x (Marshmallow) resuelve el problema del modo avión permitiendo suprimir del menú de ajustes rápidos el icono "Modo avión". Mediante el menú "Ajustes - Sistema - Configurador de IU del sistema - Ajustes rápidos", una vez localizado el icono del modo avión, es posible pulsar sobre él y arrastrarlo hasta el icono inferior "Eliminar" (ver apartado "9.10. Configurador de IU del sistema").

921. En Android 5.x no existía ninguna opción para forzar a que se solicite el código de acceso al iniciar el dispositivo, salvo si éste está cifrado, pero sí presenta el botón "Ajustes - [Personal] Seguridad - [Seguridad de la pantalla] - Bloquear al encender", que hará que el dispositivo móvil entre en la pantalla de bloqueo al reiniciarse. Esta opción no impide que se puedan recibir llamadas, mensajes, o notificaciones, lo que podría permitir a un

⁶⁸ En este escenario el dispositivo móvil está encendido, y por ejemplo, su memoria ya ha sido descifrada.

potencial atacante acceder a información desde la pantalla de desbloqueo que podría ser sensible.

922. Android 6.x introdujo una opción denominada "Inicio seguro" bajo el menú "Ajustes - [Personal] Seguridad - [Seguridad del dispositivo] Bloqueo de pantalla - [Tipo de bloqueo] - Inicio seguro". Si se activa esta opción, se forzará al usuario a introducir el código de acceso (PIN, patrón o contraseña) definido como mecanismo de bloqueo antes de iniciar el dispositivo móvil, de modo que, si no se introduce, no será posible recibir llamadas, mensajes o notificaciones, protegiendo así la información sensible del usuario. Se recomienda activar esta opción para evitar la exposición de la información sensible en caso de pérdida o robo del dispositivo. Esta opción es independiente del estado de cifrado del dispositivo, por lo que se puede utilizar tanto con él cifrado como sin cifrar.

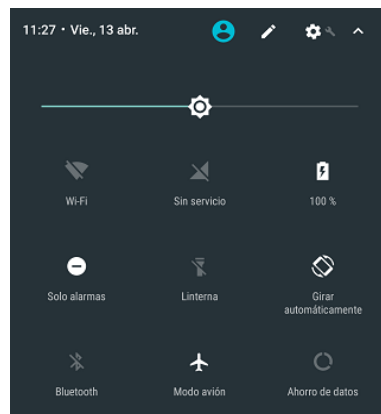


9.12. MENÚ DE AJUSTES RÁPIDOS Y BARRA DE NOTIFICACIONES

923. El menú de ajustes de configuración rápidos (ajustes rápidos por abreviar), es un menú que permite realizar determinadas configuraciones desde la pantalla actual sin tener que recorrer todo el camino desde el menú principal de "Ajustes" y los correspondientes submenús que dan acceso a una opción determinada, de forma similar a un acceso directo.
924. Si el dispositivo móvil tiene la pantalla bloqueada, este menú se mostrará al deslizar uno o dos dedos desde la parte superior de la pantalla hacia abajo.
925. Si el dispositivo móvil tiene la pantalla desbloqueada, al deslizar sólo con un dedo se mostrará (además de las notificaciones activas), un pequeño menú con iconos en blanco o en gris (según si la funcionalidad asociada a ellos está activa o no en ese momento) que corresponden a los seis primeros elementos del menú de ajustes rápidos. Para acceder a todos los ajustes

rápidos, se puede, bien deslizar hacia abajo con dos dedos desde la parte superior de la pantalla, bien deslizar con un solo dedo en dos ocasiones.

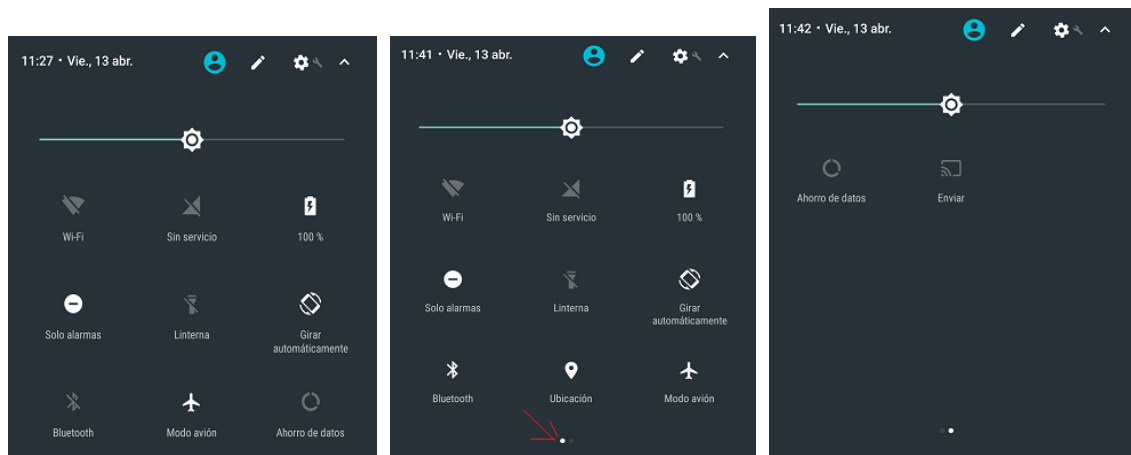
926. Los ajustes rápidos activos por defecto en Android 7.1.2 son:



927. En Android 6.x, el menú de ajustes rápidos muestra diferentes opciones según si se invoca desde la pantalla de bloqueo o desde la pantalla con el dispositivo móvil desbloqueado. Por ejemplo, con la pantalla bloqueada, no se mostrarán el botón "Ubicación", ni tampoco "Enviar" (opción que permite compartir la pantalla actual con otro dispositivo a través de Wi-Fi Direct).

928. Sin embargo, en Android 7.x, todos los ajustes rápidos habilitados en el menú se muestran en ambas pantallas. Ahora bien, sólo algunos podrán ejecutarse sin necesidad de introducir el código de acceso, y, si se selecciona alguno que requiere autenticación (como "Ubicación" o "Enviar"), el sistema solicitará dicho código de acceso al usuario.

929. Los ajustes se muestran en color blanco si están activados y atenuados (o en color gris) si están desactivados (en caso de que el número de ajustes rápidos habilitados sea superior a 9, se mostrarán dos puntos en la parte inferior del menú y, al desplazar el dedo a izquierdas, se presentarán el resto de ajustes activos):

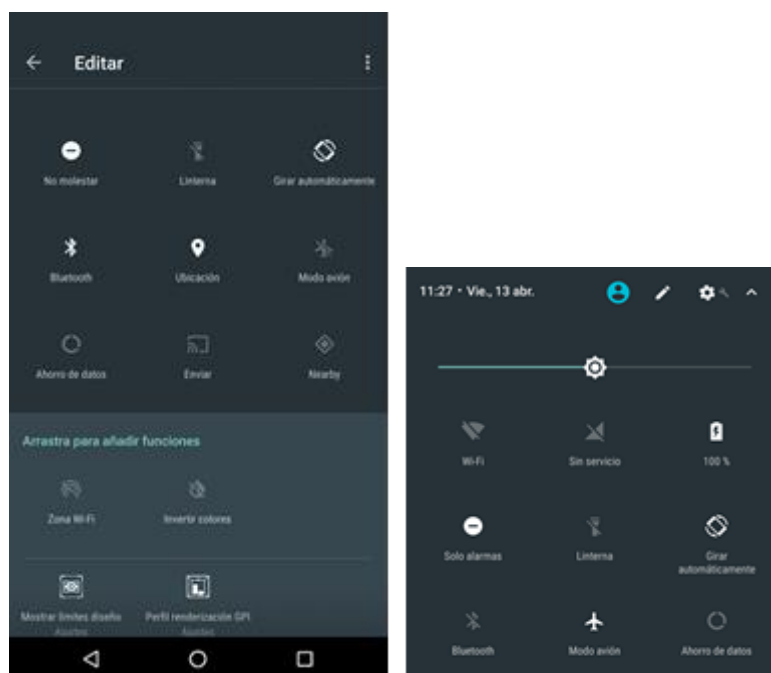


930. Los ajustes rápidos existentes en el menú de Android 7.x son los siguientes:

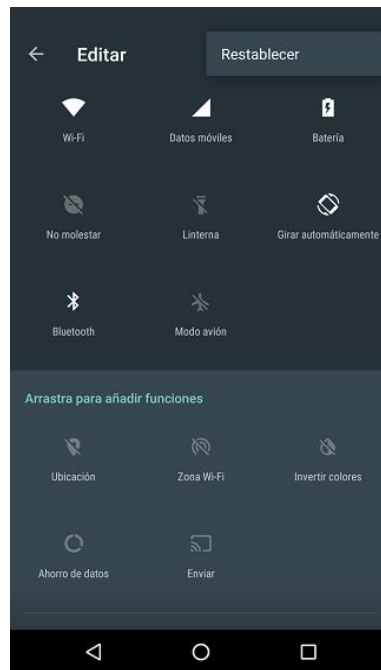
- Fecha y hora.
- Wi-Fi, red móvil y Bluetooth: además del nombre de la conexión o red empleada, se representa el estado de la transmisión de datos:

- Una flecha hacia arriba indica que el dispositivo está transmitiendo datos hacia Internet.
 - Una flecha hacia abajo muestra que se están transmitiendo datos desde Internet hacia el dispositivo.
 - Un signo de exclamación indica que hay un problema con la conexión.
- Batería: permite consultar el estado actual de carga de la batería.
- No molestar: activa un modo en el que ninguna aplicación emitirá sonidos o vibraciones.
- Linterna: encenderá el flash de la cámara del dispositivo móvil para ser usado como linterna.
- Girar automáticamente: permite habilitar la autorrotación de la pantalla del dispositivo según su orientación (si se fija a un valor *horizontal* o *vertical*, se mostrará dicho valor).
- Modo avión: habilita/deshabilita todas las conexiones inalámbricas del dispositivo.
- Ubicación: permite activar los servicios de localización geográfica (sólo se mostrará si el dispositivo está desbloqueado).
- Enviar: permite compartir la pantalla actual con otro dispositivo a través de Wi-Fi Direct (por ejemplo, con un dispositivo ChromeCast). Sólo se mostrará con la pantalla desbloqueada.
- Invertir colores: permite invertir los tonos negros y blancos de la pantalla. Para que aparezca, requiere que la opción "Ajustes - [Sistema] Accesibilidad - [Pantalla] Invertir colores" haya sido habilitada previamente.
- Zona Wi-Fi: sólo se mostrará si se ha configurado una zona Wi-Fi para compartir la conexión del dispositivo móvil con otros dispositivos a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Anclaje a red y zona Wi-Fi".
- Mostrar límites de diseño: si se activa, mostrará todos los bordes de cada elemento o menú que se encuentre en la pantalla.
- Perfil renderización GPU: para procesar todos los gráficos de aplicación en 2D a fin de mejorar el rendimiento (no siempre resulta eficaz).
- Nearby: servicio de Android basado en la ubicación que Google comenzó a desarrollar en el año 2014 [Ref.- 274]; su objetivo es localizar sitios, servicios y dispositivos con los que el terminal móvil puede interactuar que estén en un radio de acción de unos 30 m. Cuando el servicio localiza un elemento en este radio que también soporta el servicio, lanzará una notificación con recomendaciones sobre acciones que el dispositivo podrá realizar. Por ejemplo, al entrar en un museo y detectar un beacon (consultar apartado "15.9. Google Nearby"), podrá recomendar al usuario descargar una app que facilite el recorrido. En términos de seguridad, no se recomienda habilitar ni hacer uso de esta funcionalidad.

931. Hasta Android 6.x no era posible personalizar qué ajustes se incluían como parte del menú de ajustes rápidos, y tampoco se permitía deshabilitarlos de forma que no apareciesen en la pantalla de bloqueo, por lo que no había forma de evitar que un potencial atacante pudiera acceder a ellos, un comportamiento muy relevante desde el punto de vista de seguridad.
932. Android 6.x introdujo como una de sus novedades el menú "Configurador de IU del sistema" (ver apartado "9.10. Configurador de IU del sistema"), el cual permite realizar personalizaciones sobre la barra de estado y el menú de ajustes rápidos. En Android 7.x, la personalización de los elementos del menú de ajustes rápidos sale del configurador de IU del sistema y se integra como parte del propio menú de ajustes rápidos, a través del icono de "lápiz", cuya pulsación da acceso a la configuración del menú.

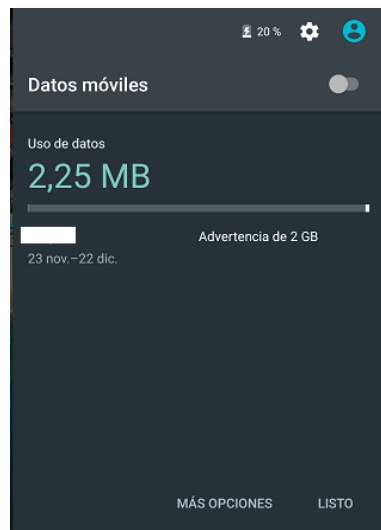


933. Al igual que ocurría en versiones anteriores de Android, no se puede definir qué elementos estarán disponibles únicamente en el menú de ajustes rápidos con la pantalla bloqueada y cuáles se podrán acceder sólo con dispositivo desbloqueado.
934. Los ajustes rápidos activos por defecto en Android 7.x no requieren tener el dispositivo desbloqueado para lanzarse. Del resto de ajustes no activos por defecto, todos salvo "Invertir colores" solicitarán el código de acceso al dispositivo si el mismo no está desbloqueado.
935. Se puede restaurar el menú de ajustes rápidos al que estaba disponible por defecto inicialmente (antes de realizar ninguna personalización) desde el menú "<Lápiz> - Editar - [...] Restablecer".



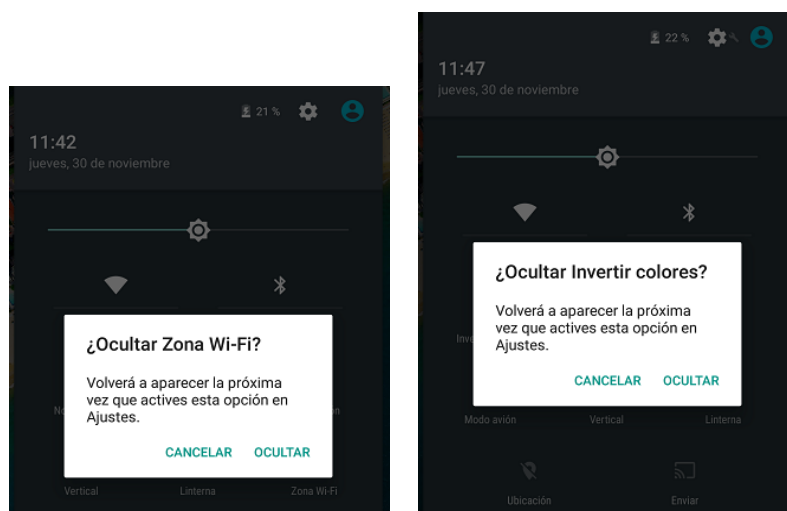
936. Con la pantalla bloqueada, además de los ajustes ya existentes en Android 4.x (Wi-Fi, Bluetooth, y girar automáticamente), en Android 5.x se añadió una opción de "Linterna" que persiste en Android 7.x y que permite utilizar el flash de la cámara del dispositivo móvil como si fuera una linterna.

937. Adicionalmente, se muestra también un icono con el nombre del proveedor de servicios de la red de datos móviles que permite consultar el consumo actual de datos para el período mostrado y si hay alguna advertencia sobre el consumo, previamente configurada en el menú "Ajustes - [Conexiones inalámbricas y redes] Uso de datos".



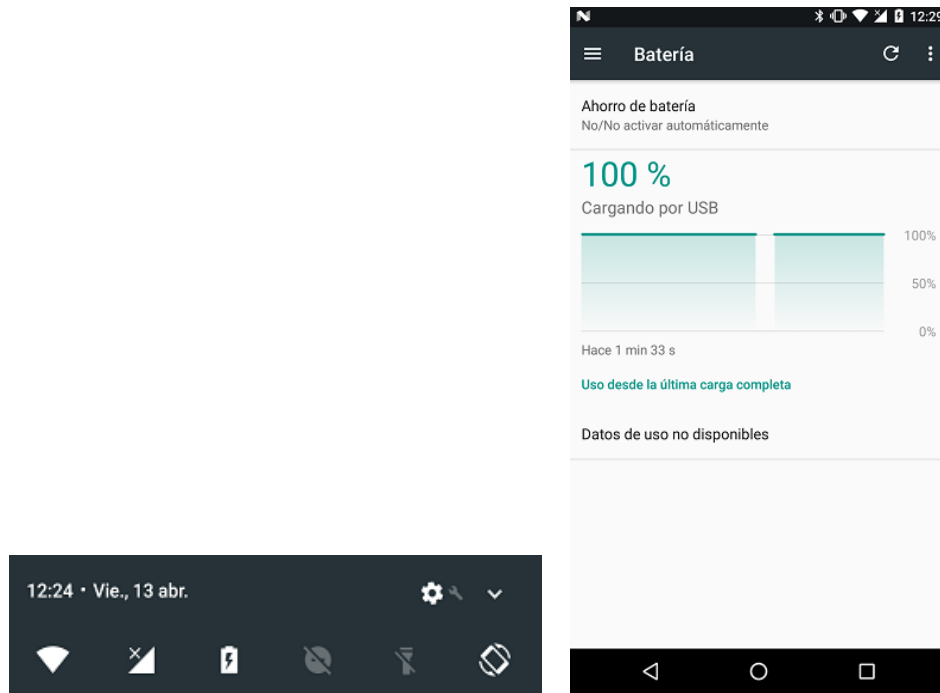
938. Con la pantalla desbloqueada, desde Android 5.x en adelante, se incorpora además la opción "Ubicación" (para activar o desactivar el servicio de ubicación, de forma equivalente al menú "Ajustes - [Personal] Ubicación") y la opción "Enviar" para compartir la pantalla (opción que permite compartir la pantalla actual con otro dispositivo a través de Wi-Fi Direct, tal como se ha descrito previamente).

939. Desde Android 6.x, a través del icono de ajustes rápidos del interfaz Wi-Fi, se permite, además de activar o desactivar el interfaz Wi-Fi del dispositivo móvil, elegir mediante una sola pulsación la red Wi-Fi a la que conectarse. Si bien el modo de activación funciona con la pantalla bloqueada, la opción de cambiar de red Wi-Fi solicitará autenticación al usuario, a menos que las credenciales de conexión a la red Wi-Fi que se seleccione hayan sido guardadas previamente. Este mismo comportamiento se exhibe con el ajuste rápido del interfaz Bluetooth.
940. Deja de estar disponible la posibilidad que introdujo Android 5.1 para eliminar fácilmente de la barra de ajustes rápidos las opciones dinámicas "Zona Wi-Fi" e "Invertir colores"⁶⁹, mediante una pulsación prolongada sobre los iconos correspondientes y seleccionando "Ocultar" en el menú emergente (las imágenes corresponden a Android 6.x, ya que en Android 7.x no se presenta esta posibilidad).

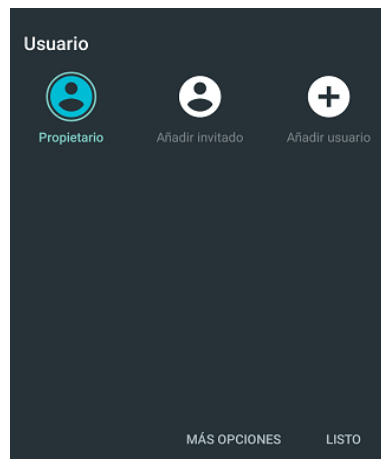


941. Adicionalmente, en la parte superior del menú de ajustes rápidos (en el área que ocupa la barra de estado) se incluye el acceso directo al menú "Ajustes - Batería" y un acceso al menú principal de "Ajustes" que, al seleccionarlo, abre dicho menú (en caso de que el dispositivo móvil esté bloqueado, se solicitará el código de acceso para abrir el menú completo).

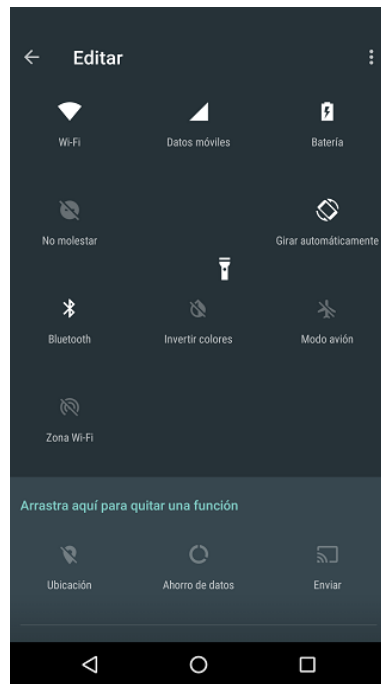
⁶⁹ La opción "Invertir colores" se activa desde "Ajustes - [Sistema] Accesibilidad - [Pantalla] Invertir colores".



942. También en la parte superior del menú de ajustes rápidos (en el área que ocupa la barra de estado) se muestra un acceso al perfil de usuario activo actualmente en el dispositivo móvil, y, seleccionándolo, se presenta la ventana con los usuarios existentes en el dispositivo, permitiéndose poder cambiar a cualquiera de ellos (sin solicitarse ningún tipo de código de acceso para realizar el cambio; sin embargo, sí se solicitará el código de acceso si el usuario al que se cambia ha definido un código de acceso propio). Para más información sobre el perfil multi-usuario, consultar el apartado "9.19. Múltiples perfiles de usuario".



943. Para añadir o quitar elementos del menú de ajustes rápidos, se ha de pulsar en el icono correspondiente de forma prolongada y arrastrarlo hasta el menú inferior, en el que se encuentran las opciones que no se han añadido al menú:

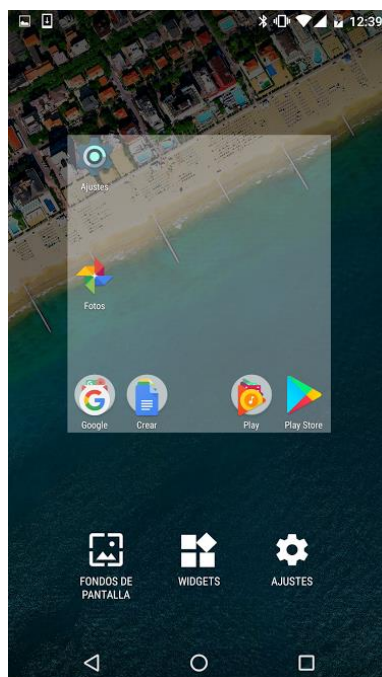


944. Desde el menú de edición de los ajustes rápidos se permite reordenar los iconos sin más que seleccionarlos y arrastrarlos manteniendo la pulsación sobre ellos hasta la posición en la que se desee presentarlos.
945. Se recomienda eliminar el icono de modo avión del menú de ajustes rápidos para evitar que el dispositivo móvil no pueda ser gestionado remotamente en caso de pérdida o robo.
946. También se recomienda eliminar del menú de ajustes rápidos tanto el acceso al interfaz Wi-Fi como a la red de datos móviles, especialmente si se desea habilitar las capacidades de gestión remota del dispositivo (útiles en caso de pérdida o robo; ver apartado "8.1. Gestión remota de dispositivos Android por parte del usuario").
947. De igual modo, se aconseja eliminar el icono de Bluetooth, que, al ser desactivado por un tercero malintencionado, podría impedir que el dispositivo pudiera ser localizado a través de, por ejemplo, un Smart Watch actualmente emparejado con él. Por el contrario, su activación malintencionada podría permitir la realización de ataques inalámbricos a través del interfaz Bluetooth del dispositivo móvil.
948. En términos generales, lo importante es entender que algunos ajustes rápidos permiten activar y desactivar controles críticos del dispositivo móvil sin necesidad de autenticación. Por este motivo, se recomienda minimizar el número de controles disponibles desde la pantalla de bloqueo.

9.13. PANTALLA DE DESBLOQUEO: WIDGETS Y CARPETAS

949. Android, desde la versión 4.2⁷⁰, permite añadir *widgets* (o complementos) que facilitan mostrar información y contenidos en la pantalla de desbloqueo, especialmente cuando ésta se encuentra bloqueada.

950. A diferencia de Android 4.x (o versiones previas), en el que la configuración de los *widgets* se encontraba dentro del menú "Ajustes [Personal] - Seguridad [Seguridad de la Pantalla] - Habilitar widgets", a partir de Android 5.x (Lollipop) el acceso a la configuración de *widgets* se realiza con una pulsación mantenida en la pantalla de inicio:



951. Adicionalmente, a partir de Android 5.x los *widgets* no pueden ser añadidos a la pantalla de desbloqueo, reduciendo notablemente el riesgo de seguridad asociado a la revelación de información sensible a través de dicha pantalla. En Android 5.x y versiones posteriores, los *widgets* permiten disponer de accesos directos y de información en la(s) pantalla(s) principal(es) o de inicio de Android, una vez el dispositivo ha sido desbloqueado conociendo el código de acceso [Ref.- 164].

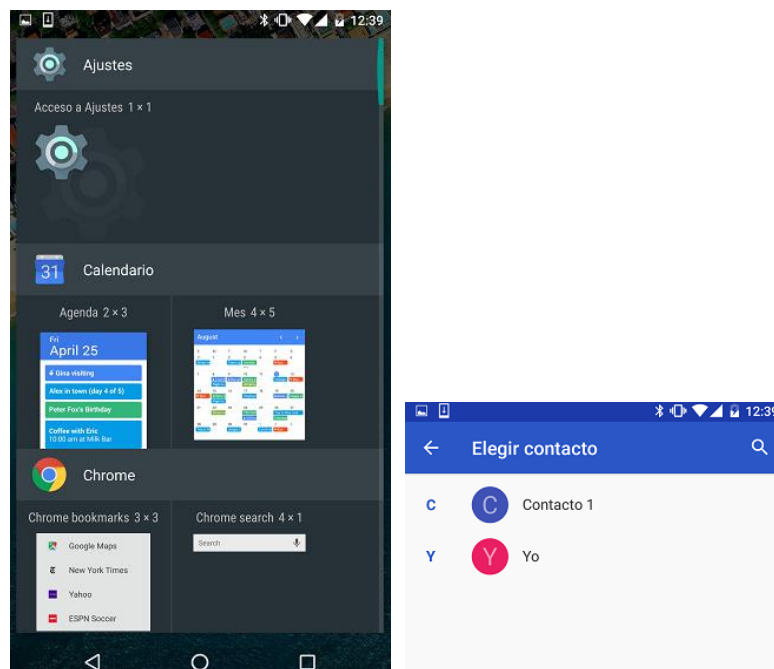
952. Los *widgets* representan normalmente versiones reducidas de ciertas apps, y permiten mostrar la información más relevante de las mismas, como por ejemplo, mensajes de e-mail, citas de calendario o información meteorológica. Se dispone también de *widgets* personalizados de apps de terceros [Ref.- 210].

953. Por defecto en la pantalla de desbloqueo de Android 7.x se muestra únicamente el *widget* asociado al reloj digital.

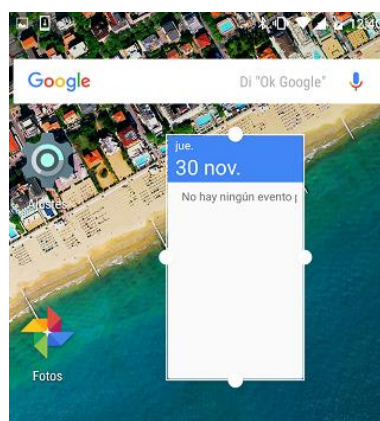
954. En Android 7.x es posible añadir nuevos *widgets* a la(s) pantalla(s) de inicio mediante el siguiente procedimiento:

⁷⁰ <http://developer.android.com/about/versions/android-4.2.html#Lockscreen>

- Pulsar un espacio vacío en la pantalla de inicio (o en cualquiera de las pantallas de inicio, en caso de disponer de varias) durante 2 segundos y marcar la opción inferior denominada "Widgets". Con ello se desplegarán todos los widgets disponibles.
- Mantener pulsado el *widget* seleccionado. El dispositivo móvil pasará a mostrar la(s) pantalla(s) de inicio de Android.
- Deslizar el *widget* a la ubicación deseada y soltar el *widget* levantando el dedo. Si el *widget* admite alguna personalización, como por ejemplo gestionar una llamada directa, al soltarlo en la pantalla de inicio que corresponda se abrirá el menú de opciones. En el caso de un acceso directo a una llamada, se mostrará un menú para elegir el contacto.



955. Para cambiar el tamaño de un *widget*, se ha de mantener pulsado el *widget* y levantar el dedo. Si se puede ajustar el tamaño, aparecerá un contorno con puntos en los laterales, y, arrastrando los puntos, se actualizará el tamaño del *widget*. Para terminar, basta con seleccionar cualquier espacio fuera del *widget*.



956. Los *widgets* pueden ser eliminados de la pantalla de inicio mediante una pulsación prolongada sobre el *widget*, y arrastrándolo hacia el texto "X Quitar" existente en la parte superior de la pantalla.
957. Desde la(s) pantalla(s) principal(es) o de inicio, es posible organizar los diferentes iconos y elementos mediante una pulsación larga sobre el icono en cuestión, lo cual permite arrastrarlo y desplazarlo a otra ubicación (incluyendo otro escritorio) y crear grupos de iconos (en Android, se denominan "carpetas"). Por defecto, Android 6.x ya organiza determinadas aplicaciones en carpetas, como "Play" (que contiene todas las apps de "Play", o "Google", que engloba las principales apps y servicios de Google, como Gmail, Maps y Fotos).
958. Para crear una carpeta, se pulsará de forma mantenida sobre un icono y se arrastrará el mismo, desplazándolo, hasta superponerlo sobre el icono o carpeta ya existente al que se desea añadirlo. Una carpeta se representa mediante un círculo que tiene destacado al icono inicial sobre el que se comenzaron a añadir los restantes.
959. Para desplegar una carpeta y seleccionar una de las aplicaciones contenidas en ella, se seleccionará brevemente el icono de la carpeta, con lo que se mostrarán todas las aplicaciones que lo integran.



960. Por defecto, una carpeta creada por el usuario no recibe nombre. Si se desea referenciarla de alguna forma, se realizará una pulsación corta sobre la carpeta y se pondrá el dedo sobre el campo "Carpeta sin nombre", con lo cual aparecerá el teclado y se podrá seleccionar el nombre deseado. Este procedimiento también es válido para cambiar el nombre previamente dado a una carpeta. Por motivos de seguridad y privacidad, se recomienda no dar a las carpetas que contengan aplicaciones sensibles nombres demasiado explícitos, como "Bancos", "Médicos", "Contraseñas"..., que podrían dar idea

a un potencial atacante con acceso visual a la pantalla sobre el tipo de aplicaciones instaladas.

- 961.Desde el punto de vista de seguridad, en el caso de querer hacer uso de *widgets*, desde Android 5.x (y vigente en Android 7.x) no existe la posibilidad de seleccionar qué *widgets* se incluirán en la pantalla de desbloqueo, y qué tipo de información, potencialmente sensible, es mostrada por estos (como sí ocurría en versiones previas de Android). En su lugar, la información disponible por defecto (y que puede ser personalizada) en la pantalla de desbloqueo se detalla en el apartado "9.6. Pantalla de desbloqueo: Previsualización de datos".

9.14. PANTALLA DE DESBLOQUEO: CÁMARA

- 962.Android, desde la versión 4.0 (*Ice Cream Sandwich*), permite hacer uso de la cámara (para hacer fotografías o grabar vídeos) desde la pantalla de desbloqueo sin necesidad de desbloquearla mediante el código de acceso.
- 963.El acceso a la cámara desde la pantalla de desbloqueo no está disponible en las tabletas basadas en Android, únicamente en los *smartphones*.
- 964.El icono de acceso directo a la cámara desde la pantalla de desbloqueo está disponible en la parte inferior derecha, mediante un icono de un cámara. Si se desplaza dicho icono (o la propia pantalla de desbloqueo) hacia la izquierda, se accede a la cámara, disponiendo de control completo sobre la misma y sus opciones de configuración:

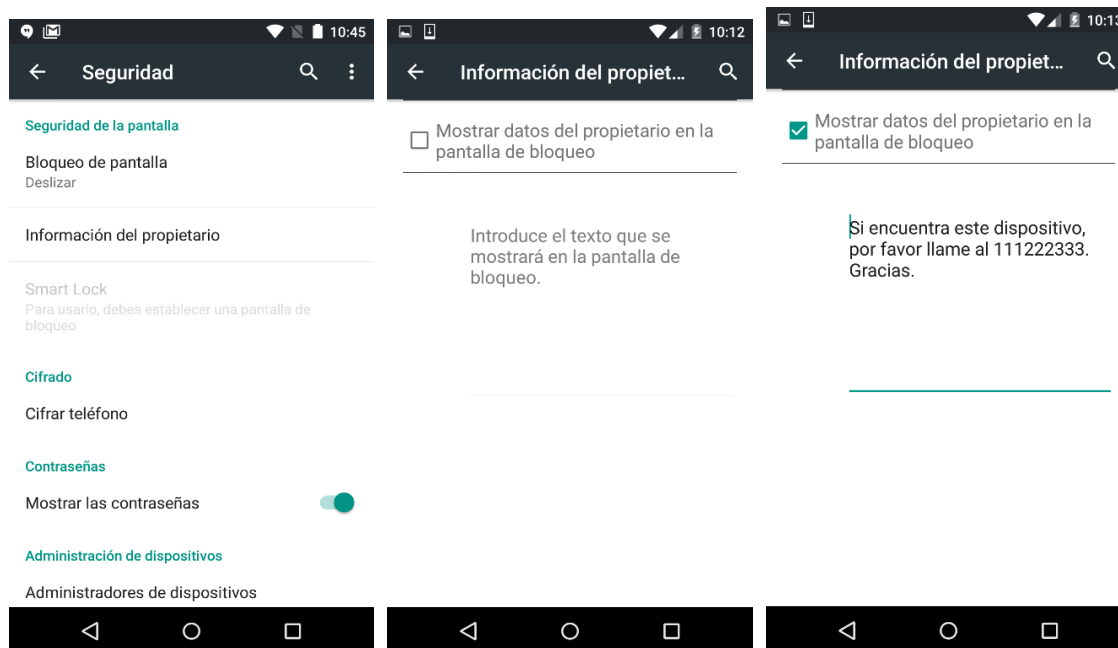


- 965.Esta funcionalidad tiene implicaciones relevantes desde el punto de vista de seguridad, ya que alguien sin autorización que obtenga acceso físico temporalmente al dispositivo móvil puede hacer fotografías o vídeos, como si estos hubieran sido realizados por el usuario del terminal.

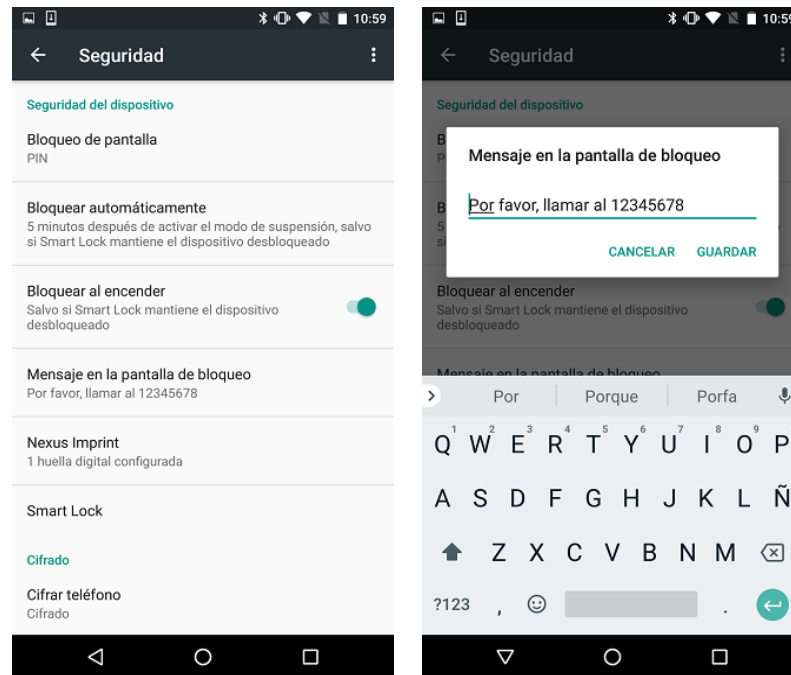
966. De hecho, se han identificado vulnerabilidades importantes que permitían que, a través del acceso a la cámara, se pudiese desbloquear la pantalla sin necesidad de introducir el código de acceso [Ref.- 163].
967. La cámara no puede ser eliminada de la pantalla de bloqueo, ya que Android no dispone de ningún ajuste de configuración con este propósito.
968. Debe tenerse en cuenta que desde la pantalla de bloqueo, respecto a la funcionalidad de la cámara, y sin conocer el código de acceso, sólo se dispone de acceso a las fotografías y vídeos que han sido realizados en la sesión actual (desplazando la pantalla de la cámara hacia la izquierda), no pudiendo visualizarse el resto de fotografías y vídeos. Ello impide que un tercero que tenga acceso físico a un dispositivo Android 7.x bloqueado pueda ver las fotografías o vídeos tomadas previamente por su propietario.

9.15. PANTALLA DE DESBLOQUEO: INFORMACIÓN DEL PROPIETARIO

969. Android permite establecer un mensaje de texto en la pantalla de desbloqueo, característica que es especialmente útil cuando la pantalla se encuentra bloqueada y se extravía el dispositivo móvil.
970. Si alguien con buenas intenciones se encuentra el dispositivo móvil, puede visualizar dicho mensaje y llevar a cabo acciones en función del mismo como, por ejemplo, contactar con su propietario.
971. Hasta Android 5.x incluido, el mensaje con información del propietario está deshabilitado, y se habilitaba a través del menú "Ajustes [Personal] - Seguridad [Seguridad de la Pantalla]" y la opción "Información del propietario":



972. Desde Android 6.x, el mensaje que se desee mostrar en la pantalla de bloques se configura a través del menú "Ajustes [Personal] - Seguridad [Seguridad del dispositivo]" y la opción "Mensaje en la pantalla de bloqueo":

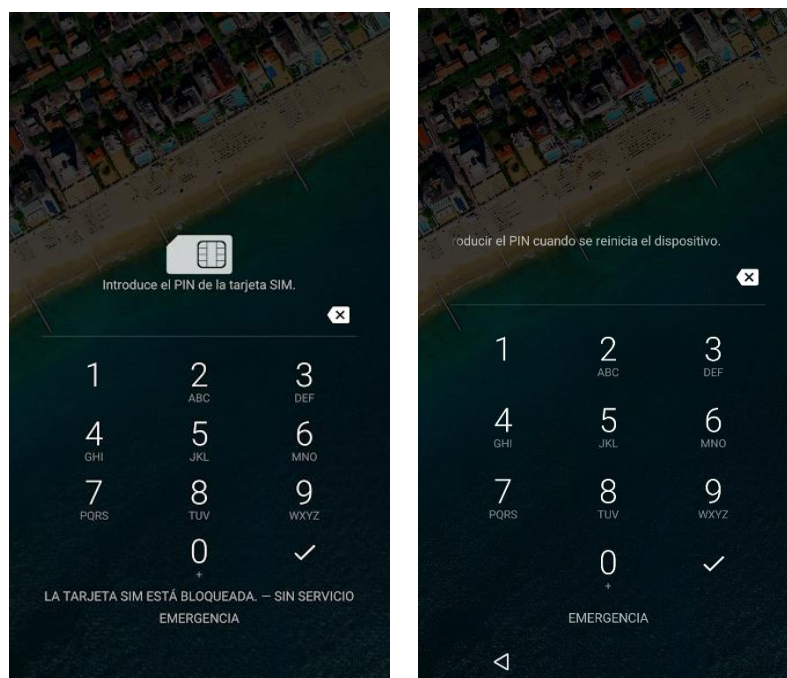


973. Se recomienda proporcionar un mensaje descriptivo que, sin desvelar información sensible, crítica, o que pueda ayudar a identificar detalles innecesarios del propietario, ayude, por ejemplo, a la recuperación del dispositivo móvil. Por ejemplo, se recomienda fijar un mensaje similar al siguiente: "Si encuentra este teléfono, por favor, llame al número +34 123456789. Muchas gracias". Si el mensaje tiene un número de caracteres que excede el tamaño del área reservada para él en la pantalla de bloqueo, se mostrará como un mensaje que se desplaza hacia la izquierda, permitiendo su lectura completa.



9.16. PRECAUCIONES Y RECOMENDACIONES ADICIONALES

974. Adicionalmente a todas las opciones de configuración e implicaciones asociadas detalladas previamente, es posible establecer éstos y otros parámetros de configuración asociados al proceso de desbloqueo del dispositivo móvil a través de las políticas de seguridad de Google G-Suite, sustituto de Google Apps desde finales de 2016 (ver apartado "8. Gestión empresarial de dispositivos móviles basados en Android").
975. Cuando se establece un código de acceso al dispositivo móvil, si el modo de depuración está habilitado (ver apartado "13.3. Opciones de desarrollo y depuración USB") es posible acceder al terminal mediante el interfaz USB, por ejemplo para capturar su pantalla, intercambiar datos, gestionar sus ficheros, instalar apps y leer datos del registro de actividades, sin necesidad de desbloquear el dispositivo móvil. Un atacante, mediante acceso físico al terminal, podría potencialmente realizar múltiples tareas de administración del dispositivo móvil sin conocer la contraseña de acceso. El impacto de las acciones que pueden ser llevadas a cabo sin conocer el código de acceso está mitigado por algunas capacidades de seguridad disponibles desde Android 4.4.2 (ver apartado "13.4. Relaciones de confianza a través de USB").
976. Se recomienda por tanto desde el punto de vista de seguridad deshabilitar el modo de depuración mediante el menú "Ajustes [Sistema] - Opciones de desarrollo", desactivando la opción "Depuración USB", y habilitarlo puntualmente cuando se requiera utilizar alguna funcionalidad de este menú.
977. Una vez fijados tanto el PIN en la tarjeta SIM como el código de acceso en el dispositivo móvil, al encender el terminal, el dispositivo móvil solicitará primero el PIN de la tarjeta SIM y después el del terminal, en este orden:



978. Pese a que la obligatoriedad de introducir dos PIN (o códigos de acceso) diferentes puede suponer una molestia para el usuario, desde el punto de

vista de seguridad se recomienda que ambos sean distintos, siendo necesario establecer los dos niveles de seguridad para que tanto la tarjeta SIM y sus servicios asociados como el dispositivo móvil y sus datos asociados estén protegidos.

979. Adicionalmente, y con el objetivo de mejorar globalmente la seguridad de los servicios asociados al dispositivo móvil, se recomienda modificar a la mayor brevedad posible el PIN de acceso al buzón de voz proporcionado por el operador de telefonía móvil al activar este servicio. Su valor debe ser diferente al valor del código de acceso del dispositivo móvil y del PIN de la tarjeta SIM.

9.17. DESBLOQUEO MEDIANTE LA CUENTA DE USUARIO DE GOOGLE

980. En versiones de Android 4.x y anteriores, existía la posibilidad de restablecer el código de acceso con la cuenta de usuario de Google o a través del administrador de dispositivos Android (ver apartado 8.1. "Gestión remota de dispositivos Android por parte del usuario").

981. Esta opción desapareció con Android 5.x (Lollipop), por lo que no se ilustra en la presente guía; si se desea conocer las características de esta funcionalidad, puede consultarse la "Guía CCN-STIC-453B - Seguridad de dispositivos móviles: Android 4.x". CCN-CERT ([Ref.- 402]).

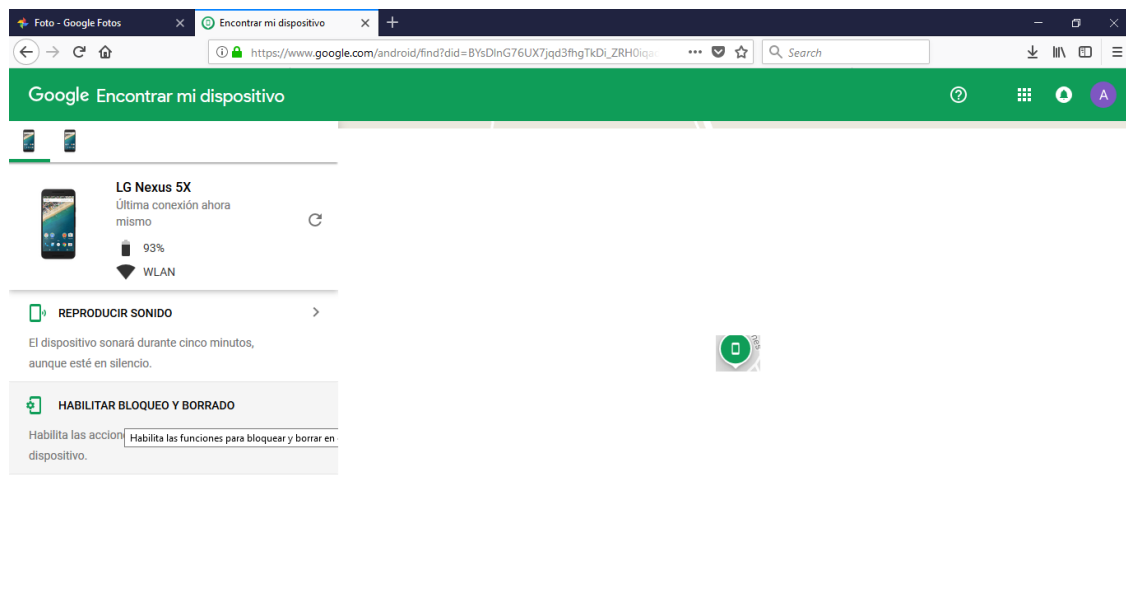
9.18. RESTAURACIÓN DEL CÓDIGO DE ACCESO

982. A través de las capacidades empresariales para la gestión de dispositivos móviles basados en Android proporcionadas por Google, y específicamente a través de la plataforma Google Apps (ver apartado 8.1. "Gestión remota de dispositivos Android por parte del usuario"), es posible llevar a cabo de forma remota la restauración (restablecimiento o reseteo) del código de acceso y desbloqueo del dispositivo móvil desde la versión 2.2 de Android, disponiendo de la app "Google Apps Device Policy"⁷¹ instalada [Ref.- 13].

983. Esta funcionalidad sólo está disponible para los clientes de Google Apps en la categoría de Negocios (Business) que utilizan G Suite (ver apartado "8. Gestión empresarial de dispositivos móviles basados en Android").

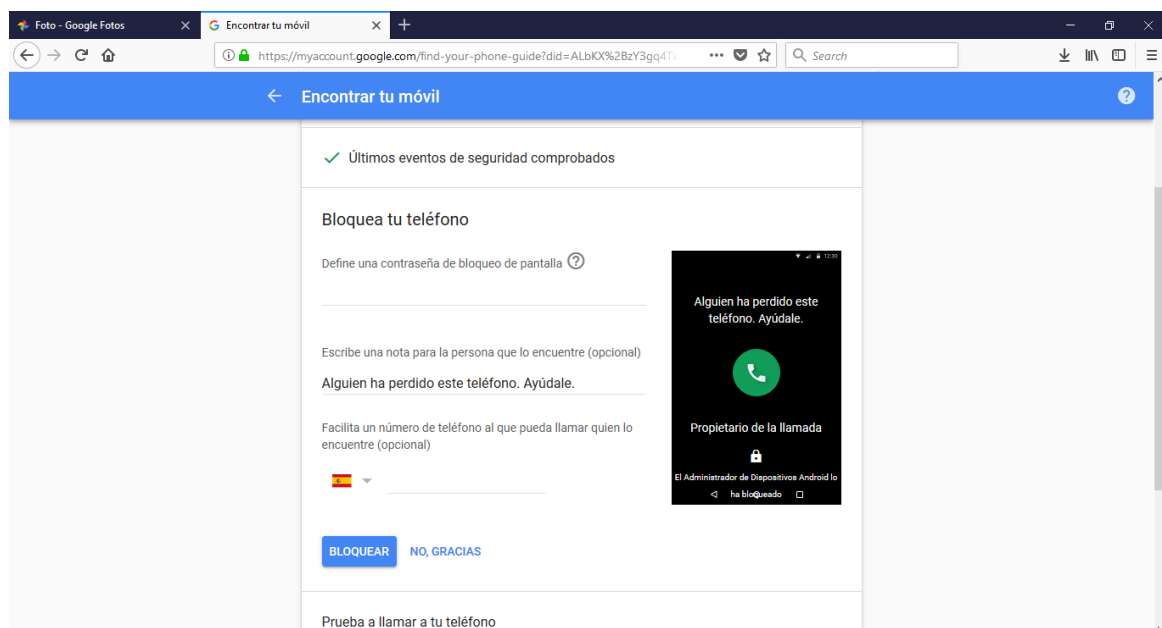
984. El resto de usuarios que disponen de una cuenta de usuario de Google individual pueden hacer uso de la funcionalidad "Encontrar mi dispositivo" (ver apartado "8.1. Gestión remota de dispositivos Android por parte del usuario"), con funcionalidad similar pero que no dispone de las capacidades para restaurar el código de acceso del dispositivo móvil, sino únicamente de "Reproducir sonido" (si se piensa que el dispositivo está en una ubicación donde la emisión de sonido pueda ayudar a localizarlo), "Habilitar bloqueo y borrado" (bloquea la pantalla y permite que se añada un mensaje personalizado) y "Borrar" (equivalente a realizar un reseteo remoto a fábrica) [Ref.- 60].

⁷¹ <https://play.google.com/store/apps/details?id=com.google.android.apps.enterprise.dmagent>



985.El servicio "Encontrar mi dispositivo", dependiendo de la versión de Android y de los servicios de Google empleada, puede aparecer como "Encontrar tu móvil", aunque se permite también localizar tabletas.

986.La opción de "Bloquear" permite, en caso de que no se hubiese configurado en el dispositivo un código de acceso previamente (escenario totalmente desaconsejado desde el punto de vista de seguridad), añadir un PIN o una contraseña al mismo.

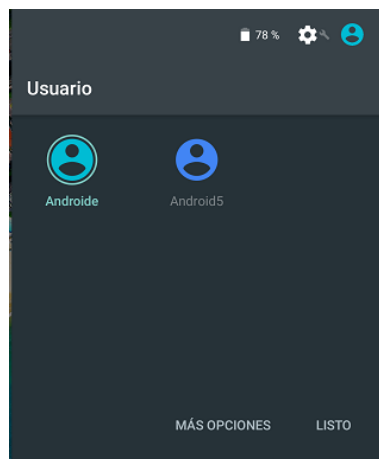


9.19. MÚLTIPLES PERFILES DE USUARIO

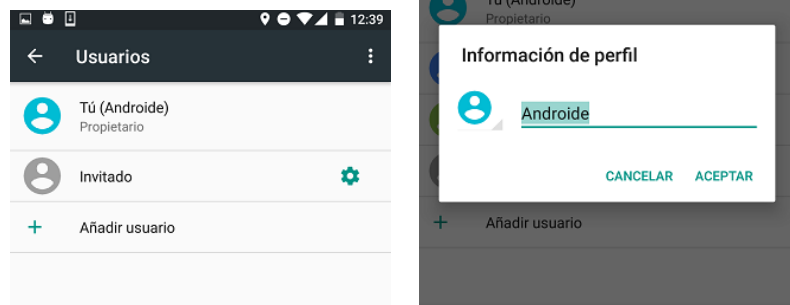
987.La versión 4.2 de Android introdujo para los dispositivos de tipo tableta la posibilidad de disponer de múltiples cuentas o perfiles de usuario, pudiendo cada usuario personalizar la pantalla de desbloqueo, el fondo de pantalla del escritorio, los complementos (*widgets*), las aplicaciones móviles (apps) y carpetas, ficheros, cuentas, y ajustes de configuración. Cada usuario dispone

de un entorno separado, incluyendo almacenamiento separado en la tarjeta de almacenamiento externa (o tarjeta SD, en caso de existir), mediante emulación. Para más detalles al respecto de la gestión de perfiles en Android 4.x, consultar la "Guía CCN-STIC-453B - Seguridad de dispositivos móviles: Android 4.x" [Ref.- 402].

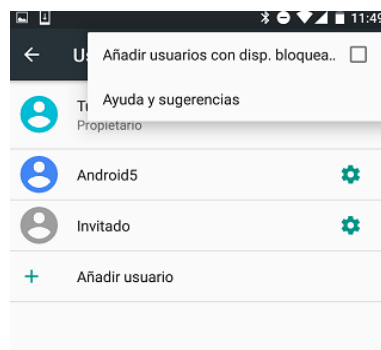
988. En Android 4.3 se introdujeron los denominados "perfiles restringidos" para los dispositivos de tipo tableta, cuyo objeto es limitar las capacidades de los distintos usuarios definidos en el dispositivo móvil [Ref.- 137].
989. Las capacidades multi-usuario en *smartphones* basados en Android 4.x sólo están disponibles si se lleva a cabo el proceso de *rooting* del dispositivo móvil, pero no oficialmente. Esta limitación se ha establecido en Android definiendo que el número máximo de usuarios en los *smartphones* es igual a uno.
990. Android 5.0 extendió la funcionalidad multi-usuario ya disponible desde Android 4.2 para tabletas a los teléfonos (no todos los teléfonos soportaban esta funcionalidad, aunque sí está disponible en los dispositivos móviles Nexus).
991. Android 7.x también mejora esta funcionalidad en tabletas, como el dispositivo Nexus 9. A diferencia de los teléfonos, las tabletas tienen la opción de crear un perfil. Los perfiles permiten limitar (o habilitar) las apps y servicios que el usuario de ese perfil puede usar. Por ejemplo, es posible limitar la búsqueda de Google, Chrome o la utilización de la cámara [Ref.- 212]. Esta funcionalidad se mantiene en Android 6.x.
992. Como conclusión fundamental de este apartado, y por las razones que se expondrán a continuación, se debe indicar que se desaconseja, desde el punto de vista de seguridad, utilizar la funcionalidad de múltiples perfiles de usuario de Android.
993. La gestión de usuarios y perfiles restringidos está disponible a través del menú "Ajustes [Dispositivo] - Usuarios", y también desde la barra de ajustes rápidos mediante el icono circular con una silueta seguida del texto "Más opciones".
994. Para ver los usuarios disponibles en un dispositivo móvil, se accederá a la barra de ajustes rápidos y se pulsará sobre el icono circular del margen superior derecho que representa un humano. Es importante reseñar que esta opción está disponible con el terminal bloqueado, por lo que cualquier persona con acceso físico al dispositivo puede ver qué usuarios existen en el mismo. Si alguno de esos usuarios no hubiese configurado su espacio, o no hubiese definido un método o código de acceso, un intruso podría tomar control de ese usuario y tener acceso a determinadas funcionalidades del dispositivo.



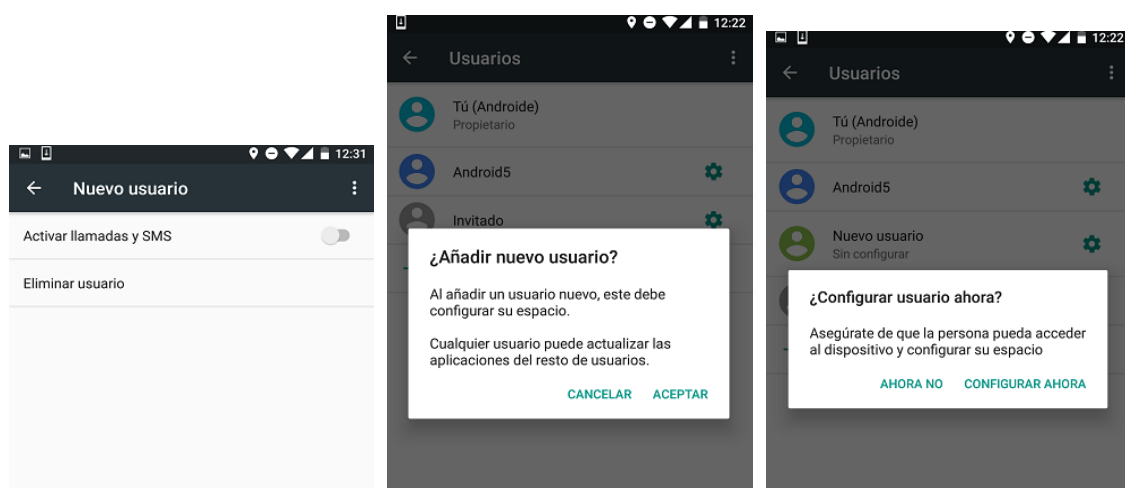
995. El propietario es el usuario que tiene control completo del dispositivo móvil y puede borrar cualquier cuenta (o perfil). El resto de usuarios pueden sólo borrar su propia información y no la de los otros usuarios (incluido, obviamente, el usuario propietario).
996. Para el usuario "invitado" cualquier personalización se almacena solo de forma temporal: cada vez que se selecciona el modo "invitado", el sistema preguntará si se desea continuar con la última sesión o volver a empezar. El objetivo del perfil de invitado es dar acceso temporal al terminal durante un muy breve periodo de tiempo.
997. Un usuario invitado no debe realizar ningún proceso de configuración inicial, cosa que sí ocurre para el resto de usuarios.
998. Al igual que ocurre con los usuarios, el perfil de invitado tiene su propio espacio en el dispositivo móvil, pero éste puede ser borrado cuando el invitado ha terminado de utilizarlo. Un invitado puede borrarse a sí mismo (y hacer así que desaparezcan los datos que se hayan podido generar durante el periodo de utilización del dispositivo móvil) desde la barra de ajustes rápidos, mediante el icono circular con una silueta seguido del texto "Eliminar invitado".
999. Por defecto, existen dos usuarios: el propietario o administrador del dispositivo (identificado como "Owner") y el Invitado. El perfil de invitado aparece directamente en la pantalla de usuarios y no se puede eliminar. El (usuario) propietario identifica al usuario inicial u original existente por defecto, y que fue empleado en el proceso de configuración inicial del dispositivo móvil.
1000. La pantalla de "Usuarios [Usuarios y perfiles]" muestra en la primera entrada al usuario actual, identificado como "Tú" y entre paréntesis el nombre del usuario (si no se definió un nombre para el usuario, aparecerá el texto "Owner") [Ref.- 211].
1001. Concretamente, el nombre por defecto es el que se proporcionó en el campo "Nombre y apellidos" durante el proceso de instalación del dispositivo móvil. Si no se rellenó este campo, se mostrará el texto "Owner". Este nombre se puede cambiar pulsando sobre la entrada del propietario ("Owner") y rellenando la "Información de perfil".



1002. El propietario puede utilizar el botón "+ Añadir usuario" para añadir nuevos usuarios. Si se quiere permitir que otros usuarios puedan añadir a su vez más usuarios, se pueden seleccionar los tres puntos situados a la derecha de "Usuarios" y marcar la opción "Añadir usuarios con disp. bloqueado". Se desaconseja por completo activar esta opción desde el punto de vista de seguridad, ya que se perdería el control a la hora de poder añadir nuevos usuarios autorizados.



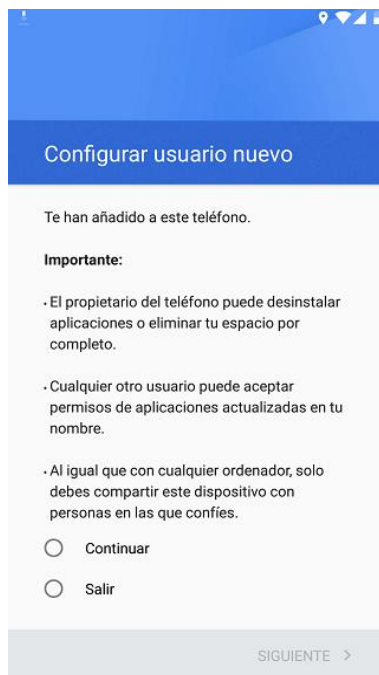
1003. Cuando se añade un nuevo usuario, se informa de que dicho usuario debe configurar su espacio de usuario, y se pregunta si se desea configurarlo en ese mismo momento, o no.



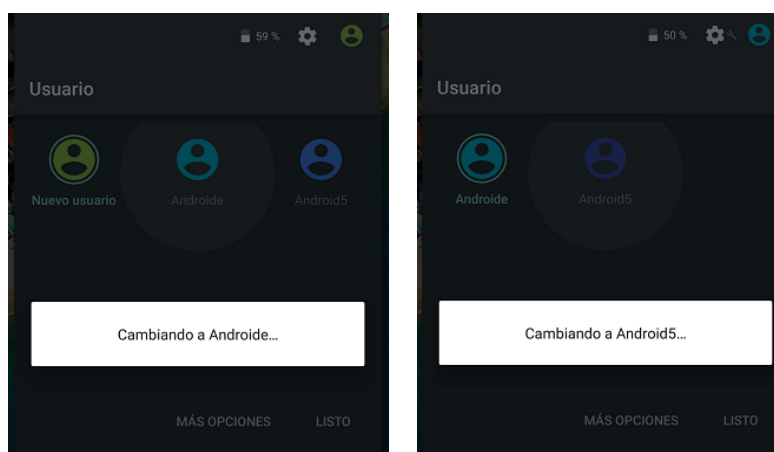
1004. Si se opta por no configurar ahora el perfil del usuario, simplemente se mostrará una entrada en la lista de usuarios con el nombre genérico "Nuevo usuario" y, entre paréntesis, se mostrará el estado: "Sin configurar". Este nombre genérico se actualizará con la información que el usuario

proporcione en el apartado "Nombre y apellidos" cuando inicie su proceso de configuración.

1005. Al pulsar sobre el nombre de un usuario de la lista disponible bajo "Ajustes [Dispositivo] - Usuarios", el terminal cambiará al perfil de ese usuario. Si no está configurado, se ofrecerá la opción de realizar la configuración de igual modo que cuando se añadió dicho usuario. Si el usuario opta por la opción "Salir" y no continúa con el proceso de configuración, dicho usuario será borrado del dispositivo y, en caso de requerirse su existencia, deberá ser creado de nuevo.



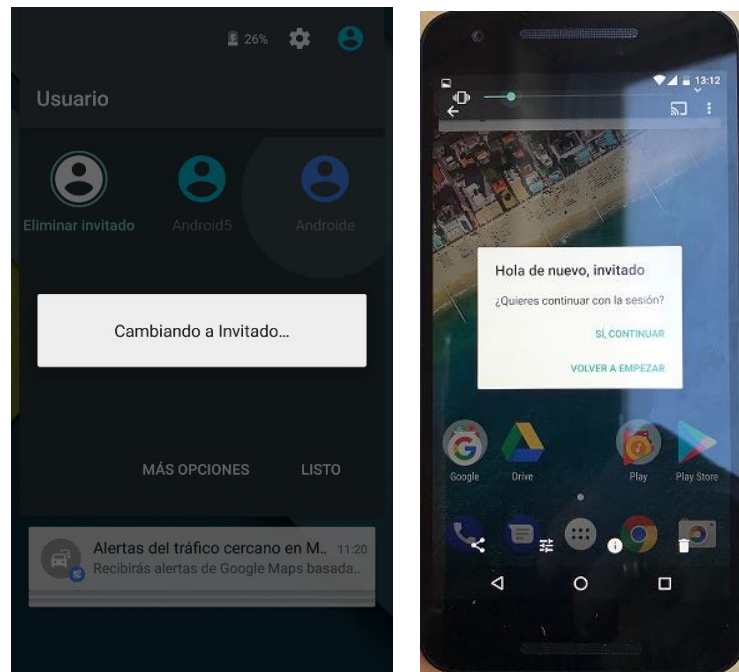
1006. La opción más rápida para cambiar entre usuarios es seleccionar el icono correspondiente a un rostro en la barra de ajustes rápidos y elegir el nombre del usuario al que se desea cambiar:



1007. En un dispositivo móvil pueden existir múltiples usuarios además del propietario, pero sólo se dispone de un usuario invitado.
1008. Para poder utilizar el perfil de invitado por primera vez, o tras resetear el perfil de invitado, es necesario entrar en "Ajustes [Dispositivo] - Usuarios" o desde la barra de ajustes rápidos (bajo el menú circular con el símbolo de

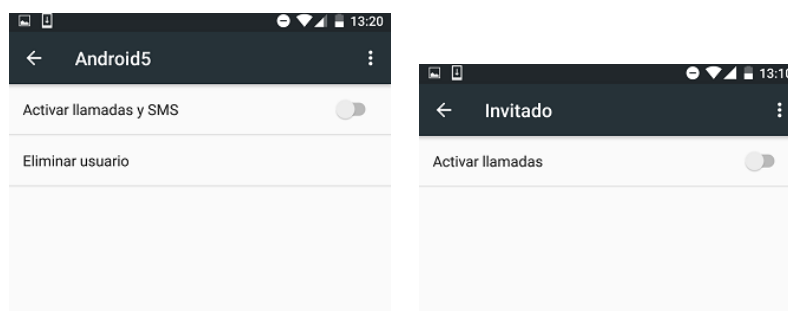
perfil seguido de "Más opciones") como usuario propietario, y seleccionar el perfil "invitado". Con esta acción, el sistema creará el perfil de invitado y cambiará a él. Hasta que no se haya creado el perfil de invitado con este método, ningún usuario distinto del propietario podrá utilizar el perfil invitado.

1009. Una vez creado el perfil invitado, el menú de usuarios de la barra de ajustes rápidos mostrará ese perfil, y cualquier usuario podrá utilizarlo.

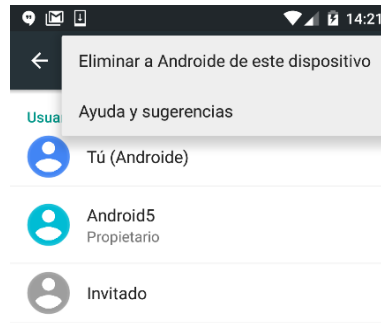


1010. Un usuario distinto del propietario puede añadir nuevas redes Wi-Fi, que estarán disponibles para todos los usuarios del dispositivo móvil, con las negativas implicaciones de seguridad asociadas.
1011. Cualquier usuario tiene acceso al menú de ajustes rápidos con la configuración por defecto de Android. Este aspecto es también muy delicado desde el punto de vista de seguridad, porque el propietario podría haber personalizado este menú para, por ejemplo, eliminar el acceso rápido al modo avión, que, sin embargo, se mostraría para cualquier otro usuario del terminal, incluido el perfil de invitado.
1012. Asimismo, los usuarios tienen acceso a muchos ajustes de configuración, si bien no a todos.
1013. Los usuarios distintos del invitado tienen su propio espacio en el dispositivo móvil con opciones de configuración, aplicaciones (apps), cuentas y pantallas de inicio personalizadas, entre otros elementos.
1014. Un usuario puede añadir su cuenta de Google al dispositivo e instalar aplicaciones desde Google Play. Las apps instaladas por un usuario son solo visibles para él (otros usuarios o el propietario no las vería). Si una app que un usuario elige instalar ya está instalada por otro usuario, se creará un espacio virtual (o *sandbox*, diferenciado) para el nuevo usuario dentro de esa aplicación (o app), de modo que cada usuario dispondrá únicamente de acceso a sus datos dentro de la app.

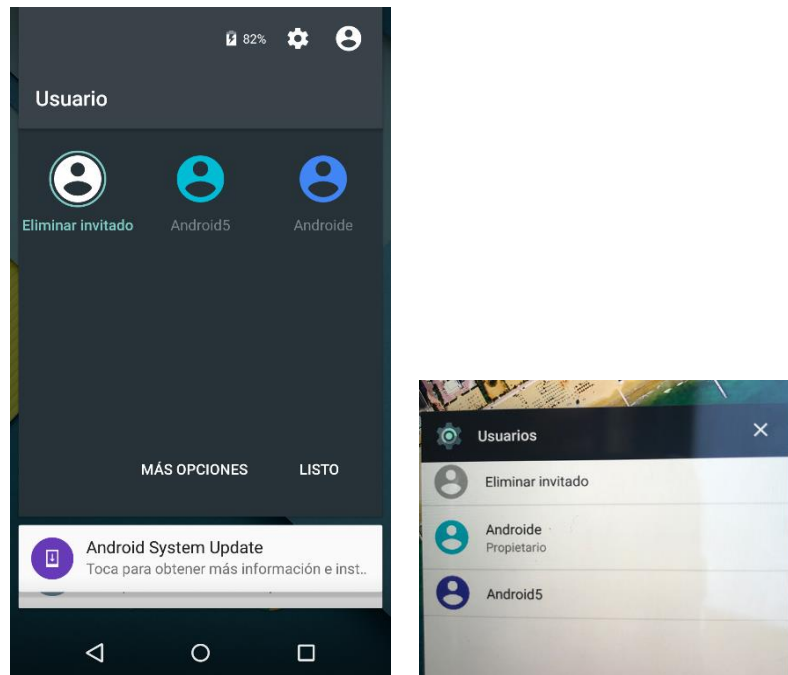
1015. Si un usuario elige actualizar una aplicación y dicha aplicación también existe para otro usuario, se actualizará la versión de la aplicación para ambos. Es decir, la actualización de apps se lleva a cabo a nivel global dentro del dispositivo móvil.
1016. Asimismo, cualquier usuario puede aceptar nuevos permisos en el proceso de actualización de apps, afectando globalmente desde el punto de vista de seguridad al dispositivo móvil y al resto de usuarios.
1017. Cualquier usuario, incluyendo el invitado, tiene opción de cambiar a otro usuario, incluido el propietario (u "Owner") del dispositivo móvil, a través de la barra de ajustes rápidos sin que se solicite ningún código para realizar esta acción (el menú de ajustes rápidos y la barra de notificaciones están disponibles tal como se detalló en el apartado "9.12. Menú de ajustes rápidos y barra de notificaciones"). Si el usuario al que se cambia tiene configurado un código de acceso, aparecerá la pantalla de bloqueo, y será necesario introducir su código. Si el usuario al que se cambia no tiene definido un código de acceso, se entrará en su espacio de usuario sin requerirse ningún tipo de autenticación.
1018. Por defecto, un usuario no tiene permisos para utilizar los servicios de telefonía. Si se desea otorgárselos, hay que seleccionar "Ajustes - [Dispositivo] Usuarios - Nombre del usuario", mediante el botón de ajustes situado a la derecha del nombre del usuario y marcar el interruptor "Activar llamadas y SMS". Para el invitado, la opción se denomina "Activar llamadas".



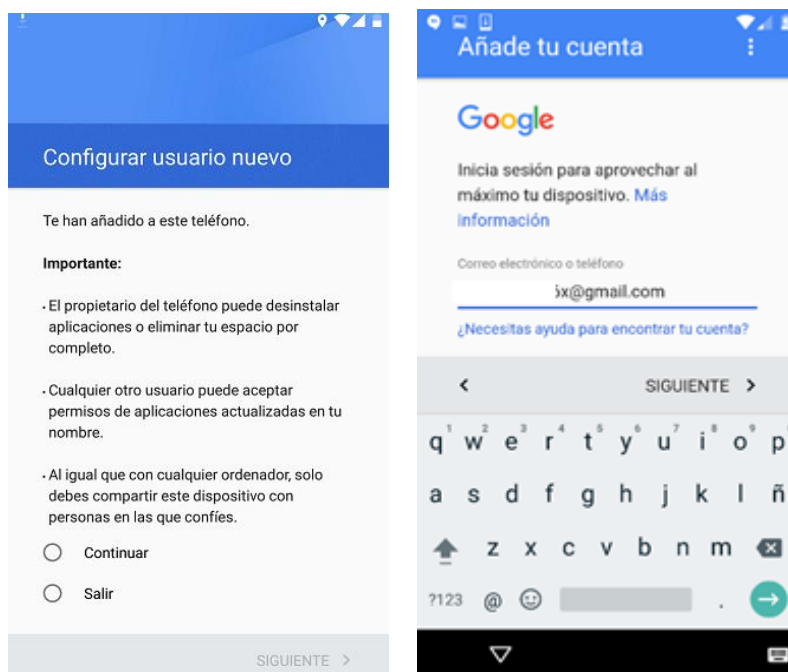
1019. Para eliminar un usuario, hay que seleccionar "Usuarios - Nombre del usuario", mediante el botón de ajustes situado a la derecha del nombre del usuario y marcar la opción "Eliminar usuario". Aparecerá una ventana de confirmación que indica que se eliminarán los datos y las aplicaciones. Si la aplicación solo ha sido instalada por ese usuario, se borrará del dispositivo móvil; si algún otro usuario la instaló, únicamente se borrará de la aplicación el espacio de datos asociado a ese usuario.
1020. Además, un usuario puede eliminarse a sí mismo sin más que entrar en su perfil, seleccionar en el símbolo de usuario de la barra de ajustes rápidos, mediante "Más Opciones - [Usuarios] tres puntos (...) - Eliminar a <nombre de usuario> de este dispositivo".



1021. Cuando se cambia de un usuario a otro, se presenta la pantalla de bloqueo del usuario destino y, tras desbloquearla, se llegará a la pantalla que este usuario tuviera activa la última vez que usó el terminal. Se mantienen por tanto todos los datos de la sesión activa.
1022. Ningún usuario distinto del propietario tiene acceso a la opción de "Copia de seguridad y restauración", por lo que no pueden realizar copias de seguridad o backups de sus datos ni acceder a eliminar los datos del dispositivo móvil (pueden eliminar los datos propios borrando su perfil).
1023. Para eliminar al usuario invitado, hay que entrar en el perfil de invitado y, desde la barra de ajustes rápidos o desde el menú "Ajustes - Usuarios [Usuarios y perfiles]", seleccionar "Eliminar invitado". El propietario del dispositivo, que sí puede borrar otros usuarios, no puede eliminar al usuario invitado. Para hacerlo, tiene que entrar en el perfil del invitado y eliminarlo desde ahí. Es importante tener esto en cuenta en caso de haber habilitado temporalmente un usuario invitado para no dejarlo disponible y activo por descuido en el dispositivo móvil.



1024. El proceso de configuración de un nuevo usuario (desde la pantalla de desbloqueo) conlleva un proceso similar al de configuración inicial del dispositivo móvil, pero en los aspectos relacionados específicamente con el nuevo usuario. Permite por ejemplo añadir (opcionalmente) la cuenta de Google del usuario (ofreciéndose la posibilidad de crear una cuenta de usuario de Google si no se dispone de una), determinar si se permitirá a las apps hacer uso de los servicios de ubicación y con qué precisión, solicitar el nombre del usuario para personalizar algunas apps, confirmar la aceptación de actualizaciones a través de los servicios de Google, y permite al usuario personalizar la pantalla de inicio.



1025. La implementación multi-usuario de Android asigna un identificador (o ID) a cada usuario (que es independiente de los identificadores o UUIDs

asignados a las apps a nivel de Linux), empezando por el identificador 0, que es asignado al usuario propietario.

1026. La información de cada usuario se almacena en el directorio `"/data/system/users/<ID>"`, donde se guardan los ajustes de configuración del usuario, las personalizaciones de su pantalla de inicio, y la lista de apps actualmente instaladas y disponibles para ese usuario.
1027. Los binarios de las apps se comparten entre los usuarios, para evitar duplicidades y un consumo innecesario de la memoria de almacenamiento, pero el directorio o *sandbox* de trabajo de la app es independiente para cada usuario, obteniéndose una copia del original en el momento de la instalación (o activación) de la app para un usuario concreto.
1028. Las capacidades multi-usuario de Android modifican el modelo de asignación de identificadores o IDs para las apps descrito en el apartado "7.3. Sandbox de ejecución de las apps", ya que al ser instalada para un usuario concreto, cada app recibe un nuevo identificador efectivo combinación del ID de usuario y del ID de la app [Ref.- 195].
1029. Desde el punto de vista empresarial o corporativo, no se recomienda la compartición de los dispositivos móviles entre distintos usuarios, por lo que se desaconseja el uso de las capacidades para disponer de múltiples perfiles de usuario en Android.
1030. En caso de identificarse escenarios donde debe compartirse el uso de un mismo dispositivo móvil Android entre distintos usuarios (por ejemplo, un terminal utilizado para un servicio de guardia con rotación entre varias personas), se recomienda llevar a cabo un análisis detallado de los requisitos, planificar adecuadamente su implementación e investigar minuciosamente las posibles implicaciones de seguridad de la puesta en marcha de las capacidades multi-usuario para la versión concreta de Android empleada.

9.20. FACTORY RESET PROTECTION (FRP)

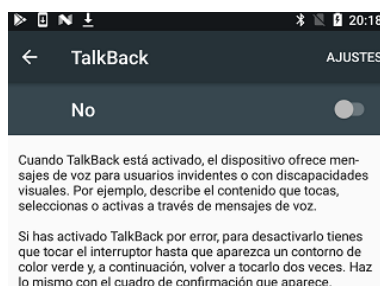
1031. El Factory Reset Protection (FRP) es un mecanismo, introducido por Google con Android 5.1, que pretende evitar que un tercero con acceso físico al dispositivo móvil efectúe un reseteo a fábrica y realice una instalación desde cero, pudiéndolo utilizar como si fuera propio (por ejemplo, tras ser robado o sustraído) [Ref.- 521].
1032. Para ello, si se intenta realizar una instalación desde cero tras un reseteo a fábrica y la protección FRP está activada, el proceso de instalación solicitará al usuario las credenciales de acceso a la cuenta de Google del propietario anterior del dispositivo. Si no se dispone de dichas credenciales, el proceso de instalación inicial no continuará.
1033. Para activar esta protección, es preciso tener definida una cuenta de Google en el dispositivo móvil. En caso de robo o pérdida, además de poder establecer un bloqueo remoto y tratar de localizar el dispositivo, según lo descrito en el apartado "8.1. Gestión remota de dispositivos Android por parte del usuario", el propietario del terminal podrá realizar un reseteo a

fábrica, el cual impedirá su reinstalación a menos que se introduzcan las credenciales de la cuenta de Google activa en el dispositivo antes de realizar el reseteo a fábrica.

1034. Para desactivar esta protección, se debe eliminar la cuenta de Google del dispositivo desde "Ajustes - Cuentas". Desde Android 6.x, en algunos dispositivos móviles se desactiva también la protección si se lleva a cabo un reseteo a fábrica legítimo desde los ajustes del dispositivo, estando desbloqueado.
1035. A lo largo de las diferentes versiones de Android han existido vulnerabilidades que han permitido saltarse esta protección, por lo que se recomienda mantener actualizada la versión de Android para evitar este tipo de debilidades [Ref.- 280].
1036. Desde el punto de vista de seguridad, se recomienda mantener activa esta opción si el dispositivo dispone de una cuenta de usuario de Google.

9.21. TALKBACK

1037. Talkback es un servicio incluido por defecto en los dispositivos Android que permite interactuar con el dispositivo móvil a través de mensajes de voz y sin necesidad de acceso visual a la pantalla [Ref.- 278].
1038. Este servicio se activa a través del menú "Ajustes - [Sistema] Accesibilidad - Talkback". Bajo este menú se encuentra un interruptor que activa el servicio, y un botón de "Ajustes" situado en el margen superior derecho a través del cual se configuran las diferentes opciones del servicio⁷²:



1039. Dado que el servicio de Talkback permite que contenidos del dispositivo se proporcionen de forma verbal, y, por tanto, puedan ser conocidos por parte de terceros sin que estos dispongan de acceso visual a la pantalla, desde el punto de vista de seguridad se desaconseja su utilización a menos que las circunstancias lo requieran.

10. CIFRADO DE DATOS EN ANDROID

10.1. CIFRADO DEL DISPOSITIVO MÓVIL

1040. Android 7.x dispone de capacidades para el cifrado de los datos almacenados en la memoria interna del dispositivo móvil de manera nativa,

⁷² Los detalles de configuración del servicio TalkBack quedan fuera del ámbito de la presente guía.

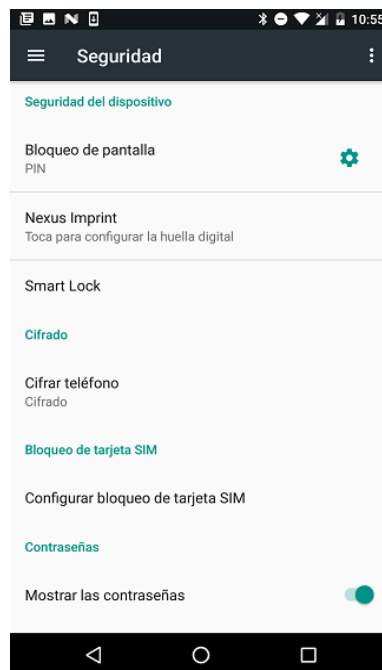
de forma que los mismos sólo se puedan acceder cuando el dispositivo está desbloqueado [Ref.- 85].

1041. Desde la versión 3.0 de Android ("Honeycomb") [Ref.- 31], orientada a dispositivos móviles de tipo tableta o *tablet*, se añadieron capacidades de cifrado nativas a los dispositivos móviles basados en Android [Ref.- 85] (empleando dm-crypt⁷³), junto a otras mejoras en las políticas de cifrado y de contraseñas (complejidad, expiración e histórico).
1042. Lógicamente, la opción de cifrado completo del dispositivo móvil también está disponible en versiones posteriores, como Android 5.x (y versiones futuras), para teléfonos móviles o *smartphones*, tanto del teléfono como (potencialmente) de la tarjeta de almacenamiento externa (en aquellos dispositivos móviles que disponen de una tarjeta SD real).
1043. Por otro lado, existen numerosas apps de cifrado en Google Play (ej. OpenKeychain: Easy PGP para Android⁷⁴) que permiten el cifrado de otros datos, como por ejemplo e-mails, notas, información confidencial o contraseñas, pero no de la totalidad del dispositivo móvil.
1044. Los mecanismos de cifrado han ido variando a lo largo de las diferentes versiones de Android, con el objetivo de proteger el cifrado de los datos frente a ataques de fuerza bruta (dado que el cifrado del dispositivo móvil está basado, o depende, del código de acceso empleado en el mecanismo de bloqueo del dispositivo).
1045. En Android 5.x (Lollipop) la contraseña de cifrado dependía sólo parcialmente del código de acceso al dispositivo móvil, interviniendo también otros elementos (que serán descritos posteriormente) en su protección [Ref.- 134].
1046. En septiembre de 2014 Google anunció que sería obligatorio que cualquier terminal con Android viniera cifrado de fábrica. Sin embargo, alegando problemas de rendimiento, finalmente el cifrado de fábrica se incluyó como "una recomendación" y no como una obligación para los demás fabricantes.
1047. Aunque Google informó públicamente de que los terminales que se fabricasen a partir de Android 5.x (Lollipop) vendrían cifrados de fábrica [Ref.- 140], los dispositivos Nexus 4, Nexus 5, Nexus 7 y Nexus 10 no se cifraron de forma predeterminada⁷⁵.
1048. A partir de Android 6.x, algunos dispositivos móviles como el empleado para la elaboración de la presente guía (Nexus 5X) sí vienen cifrados de fábrica y no es posible deshabilitar las capacidades de cifrado. Con Android 7.x, los dispositivos Nexus 5X también vienen cifrados de fábrica, no siendo posible revertir el terminal a un estado sin cifrar:

⁷³ <https://code.google.com/p/cryptsetup/wiki/DMCrypt>

⁷⁴ <https://www.openkeychain.org>

⁷⁵ <https://support.google.com/nexus/answer/2844831?hl=es>



1049. La opción disponible bajo el menú "Ajustes - [Personal] - Seguridad - [Cifrado] Cifrar teléfono" permite proceder al cifrado de la memoria del dispositivo móvil (en caso de no estar cifrado de fábrica). Una vez ha sido cifrada, se mostrará el texto "Cifrado", tal como ilustra la imagen superior. En los dispositivos Nexus que traen de fábrica la versión de Android 7.x, como el utilizado en la elaboración de la presente guía, la memoria viene ya cifrada.
1050. El proceso de cifrado es irreversible y no es posible devolver el terminal a una configuración sin cifrar. Si un dispositivo que partía de una versión de Android que no obligaba al cifrado se cifra posteriormente, la única opción posible para deshabilitar las capacidades de cifrado es restablecer el dispositivo móvil a sus ajustes de fábrica (lo que eliminará todos los datos de usuario del terminal) y restaurar posteriormente los datos de usuario desde una copia de seguridad.
1051. Al igual que Android 6.x estableció como requisito disponer de las capacidades de cifrado en todos los dispositivos móviles compatibles, Android 7.x establece como requisito disponer de soporte hardware para el almacenamiento de claves [Ref.- 706].
1052. Para obtener todos los detalles sobre el proceso de cifrado de un dispositivo móvil que parte sin cifrar, se recomienda consultar este mismo apartado en la "Guía CCN-STIC-453D - Seguridad de dispositivos móviles: Android 6.x" del CCN-CERT [Ref.- 406].
1053. Previamente a Android 7.x, cuando un dispositivo móvil cifrado arranca, tras solicitar el código de acceso al usuario durante el proceso de arranque, hacía uso de PBKDF2 (originalmente) o scrypt (posteriormente) para obtener la clave derivada (KEK, Key Encryption Key), empleada para descifrar la clave maestra (master key o DEK, Disk Encryption Key), y con esta última, poder descifrar los contenidos de la partición de datos de usuario para poder montarla y hacer uso de ella. Esto supone que, una vez arrancado el terminal

e introducido el código de acceso para descifrar los datos, las claves están en memoria, siendo utilizadas constantemente, y un tercero con acceso físico al dispositivo desbloqueado podrá también acceder a todos sus datos.

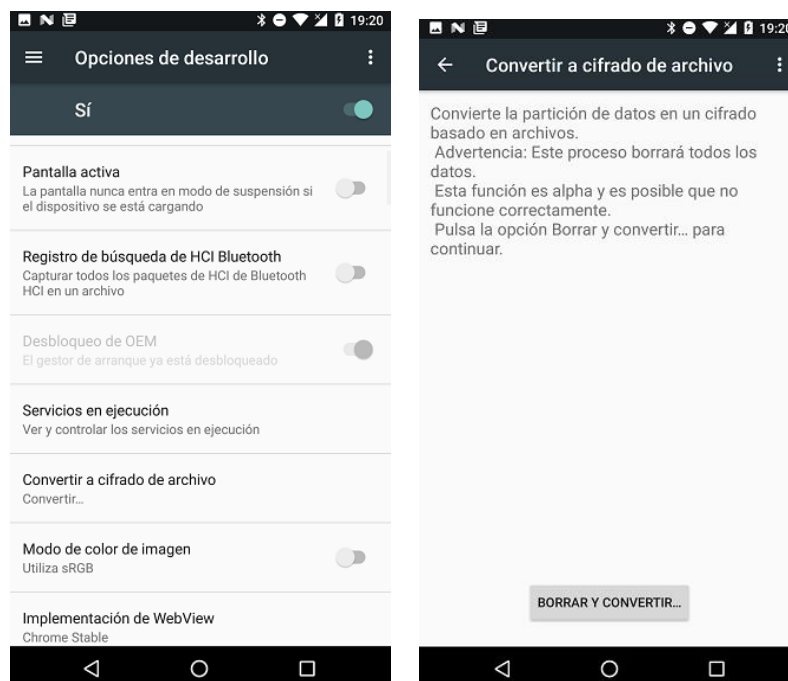
1054. El modelo tradicional de cifrado existente previamente en Android ha sido modificado en Android 7.x con nuevas capacidades. Android 7.x (Nougat) incorpora un nuevo sistema de cifrado de archivos individuales, de forma que no sea necesario cifrar el dispositivo completo con una única clave, disponiendo de la posibilidad de proteger de manera más granular tanto los datos del sistema como de las aplicaciones y del usuario.
1055. En lugar de hacer uso de FDE (*Full-Disk Encryption*) con una clave única protegida por el código de acceso del usuario para proteger todos los contenidos de la partición de datos de usuario, se hace uso de FBE (*File-Based Encryption*) [Ref.- 734], empleando diferentes claves para cifrar ficheros distintos, que pueden ser desbloqueados o accedidos independientemente.
1056. Una de las principales diferencias entre el cifrado FDE (previo a Android 7.x) es que, en él, es necesario solicitar al usuario el código de acceso durante el proceso de arranque, por lo que si el dispositivo móvil se reinicia inesperadamente en algún momento (por ejemplo, durante la noche), no estará operativo (las alarmas no funcionarán, no se podrán recibir llamadas o SMS, o no estarán disponibles los servicios de accesibilidad) hasta que el usuario sea consciente e introduzca su código de acceso para continuar con el proceso de arranque.
1057. En el caso de FBE, se pretende que la funcionalidad básica del dispositivo móvil, como la recepción de llamadas o las alarmas, esté disponible inmediatamente tras completarse el proceso de arranque. Por tanto, los dispositivos móviles que hacen uso de FBE pueden implementar las nuevas capacidades de arranque o inicio directo Direct Boot (ver apartado "10.5. Direct Boot"), que permiten al dispositivo móvil cifrado completar el proceso de arranque directamente hasta la pantalla de desbloqueo.
1058. Las apps de sistema⁷⁶ que desean ofrecer parte de su funcionalidad tras completarse el proceso de arranque del dispositivo móvil deben hacer uso de las nuevas APIs asociadas a Direct Boot (y FBE), siendo conscientes de la existencia de cifrado y pudiendo operar en un contexto limitado, incluso antes de que el usuario proporcione su código de acceso y aun manteniendo sus datos más privados y sensibles protegidos (de manera granular).
1059. En un dispositivo que hace uso de FBE, cada usuario del dispositivo dispone de dos ubicaciones disponibles para las aplicaciones en el almacenamiento:

⁷⁶ Incluyendo las apps de sistema proporcionadas por algunos fabricantes de dispositivos móviles Android.

- Credential Encrypted (CE) storage: almacenamiento basado en credenciales, disponible una vez se haya desbloqueado el dispositivo mediante el código de acceso, y que es el empleado por defecto.
 - Device Encrypted (DE) storage: almacenamiento disponible tanto durante el arranque con Direct Boot como tras el desbloqueo del dispositivo.
1060. Esta diferenciación establece una protección más allá de la introducción de un código de acceso en el arranque del dispositivo, lo cual permite un mejor manejo de los distintos usuarios (especialmente en escenarios de perfiles de trabajo). Con FBE, el área de almacenamiento de sistema, así como el área de almacenamiento de cada perfil de usuario, se cifran de manera independiente y con una mayor granularidad [Ref.- 706]. El perfil de cada usuario se cifra con una clave diferente, asociada a su código de acceso personal, por lo que los datos de un usuario no pueden ser accedidos por otro.
1061. Por tanto, cada usuario de un dispositivo multi-usuario posee dos claves propias: una DE y una CE. El usuario "0" (o propietario, empleado para realizar la configuración inicial del dispositivo) se considera especial y debe acceder al terminal en primer lugar.
1062. Las aplicaciones denominadas "Crypto-aware" (es decir, que son capaces de gestionar datos cifrados) pueden tratar con distintos usuarios si disponen del permiso `INTERACT_ACROSS_USERS`, pero sólo tendrán acceso a los directorios cifrados con la clave CE de los usuarios que han sido desbloqueados. Por el contrario, podrán interactuar sin restricciones en el almacenamiento DE, aunque corresponde a la app determinar si el estado del usuario es "desbloqueado" antes de acceder a su área DE [Ref.- 734].
1063. Por último, en entornos de trabajo que manejan perfiles de trabajo, se asigna también una clave CE y una DE al perfil, de modo que, cuando se completa el acceso al perfil de forma exitosa, el Keymaster (ejecutando en el TEE) proporcionará la clave TEE del perfil de trabajo.
1064. En resumen, con FBE, cada usuario tiene asociadas dos claves de cifrado, CE y DE, que son gestionadas mediante las capacidades hardware del TEE. La implementación FBE del AOSP requiere al menos de la versión 1.0 del Keymaster HAL (en el TEE).
1065. La funcionalidad asociada a FBE es gestionada por el demonio vold, encargado de la gestión y montaje de las particiones en Android, al que se Android 7.x añade las características de cifrado del sistema de ficheros ext4 a nivel de directorio.
1066. El cifrado de los datos personales basado en FBE incluye correo electrónico, mensajes de texto, contactos, datos de la cuenta de Google, datos de aplicaciones, fotos, contenido multimedia y descargas. Entre los datos que no se cifran está el tamaño de los archivos.
1067. El modelo de cifrado de Android no permite emplear la huella como elemento de seguridad (código de acceso) para descifrar los datos, lo cual se debe a que el reconocimiento de la huella está basado en la coincidencia

entre gran parte de los datos biométricos recolectados a través del lector de huellas y su representación almacenada durante el proceso de registro de la huella en el dispositivo móvil. Dado que este conjunto de datos puede variar, no sería posible extraer una clave concordante al 100%, requisito necesario para los algoritmos de cifrado.

1068. Los dispositivos que se actualizan a Android 7.x desde versiones anteriores de Android no serán convertidos de forma automática al modelo de cifrado FBE, ya que parten del modelo FDE. Para comprobar si el dispositivo está empleando FDE o FBE, se puede acceder al menú de establecimiento del código de acceso a través de "Ajustes - Seguridad - [Seguridad del dispositivo] Bloqueo de pantalla - [Tipo de bloqueo]". Si se presenta el menú que ofrece la posibilidad de "Requerir <código> para iniciar dispositivo", esto indica que el tipo de cifrado sigue siendo FDE.
1069. En Android 7.x queda en manos del fabricante del dispositivo móvil decidir qué modelo de cifrado será empleado, el tradicional FDE o el nuevo FBE, no siendo posible para el usuario seleccionarlo inicialmente. En algunos modelos de dispositivo móvil el usuario podrá cambiar de uno a otro.
1070. Para habilitar FBE en caso de disponer actualmente de FDE, es necesario acceder a las opciones de desarrollo "Ajustes - [Sistema] {} Opciones de desarrollo - Convertir a cifrado de archivo". Es muy importante reseñar que esta operación implica un reseteo a fábrica y todos los datos se perderán, como ilustra el mensaje de solicitud de confirmación (se solicitará introducir el código de acceso antes de proceder a la conversión):



1071. También puede cambiarse a cifrado FBE a través de ADB y fastboot (esta operación también implicará un reseteo a fábrica y la consecuente pérdida de datos):


```
$ adb reboot-bootloader  
$ fastboot --wipe-and-use-fbe
```

1072. Para que la implementación FBE pueda ser usada de manera segura, se requiere que el inicio seguro (Verified Boot) se complete de forma exitosa, con el objetivo de impedir que las credenciales de tipo DE no sean accedidas por un componente malicioso.

1073. De cara a las actualizaciones de sistema en un entorno FBE, y dado que la partición de recuperación (recovery) no tiene acceso al almacenamiento DE en la partición de datos de usuario (*userdata*), se recomienda que el dispositivo emplee el modelo "A/B Seamless updates" (ver apartado "6.5. A/B (Seamless) System updates"), puesto que la actualización OTA se realiza sin necesidad de que la partición de recuperación acceda a los datos de usuario cifrados.

1074. Si el dispositivo móvil no soporta el modelo "A/B Seamless updates", y por tanto emplea el modelo OTA tradicional en el que la partición de recuperación necesita acceder al fichero OTA en la partición de datos de usuario, se creará un directorio en lo alto de la jerarquía de la partición *userdata* que albergará los paquetes OTA, y se añadirá una regla SELinux (ver apartado "7.8. SELinux en Android") que controlará que únicamente las apps que son objeto de la actualización puedan leer o escribir en ese directorio.

10.2. ALMACENAMIENTO ADOPTABLE

1075. Android 6.x introdujo el concepto de dispositivos de almacenamiento flexibles o adoptables [Ref.- 603], es decir, dispositivos de almacenamiento externos como, por ejemplo, tarjetas SD o discos USB, que el usuario puede incorporar al dispositivo móvil, ampliando así sus capacidades de almacenamiento totales.

1076. Para la conexión de dispositivos USB, se requiere como interfaz físico un cable o un adaptador USB OTG (On-the-Go).

1077. Un dispositivo de almacenamiento adoptable puede configurarse, bien como almacenamiento portátil (para datos que pueden accederse desde distintos dispositivos móviles como, por ejemplo, contenido multimedia), o como almacenamiento interno (como complemento al almacenamiento interno del dispositivo móvil, para almacenar aplicaciones y los datos privados de éstas).

1078. Android 7.x es capaz de detectar una unidad de almacenamiento externa de forma automática, y lanzará una notificación para permitir al usuario su configuración.

1079. Los dos tipos de configuración de un dispositivo externo son *portable* (portátil) o *internal* (interno). Los dispositivos de almacenamiento que se definan como internos se formatearán utilizando ext4 como sistema de ficheros, mientras que los externos se formatearán con FAT32.

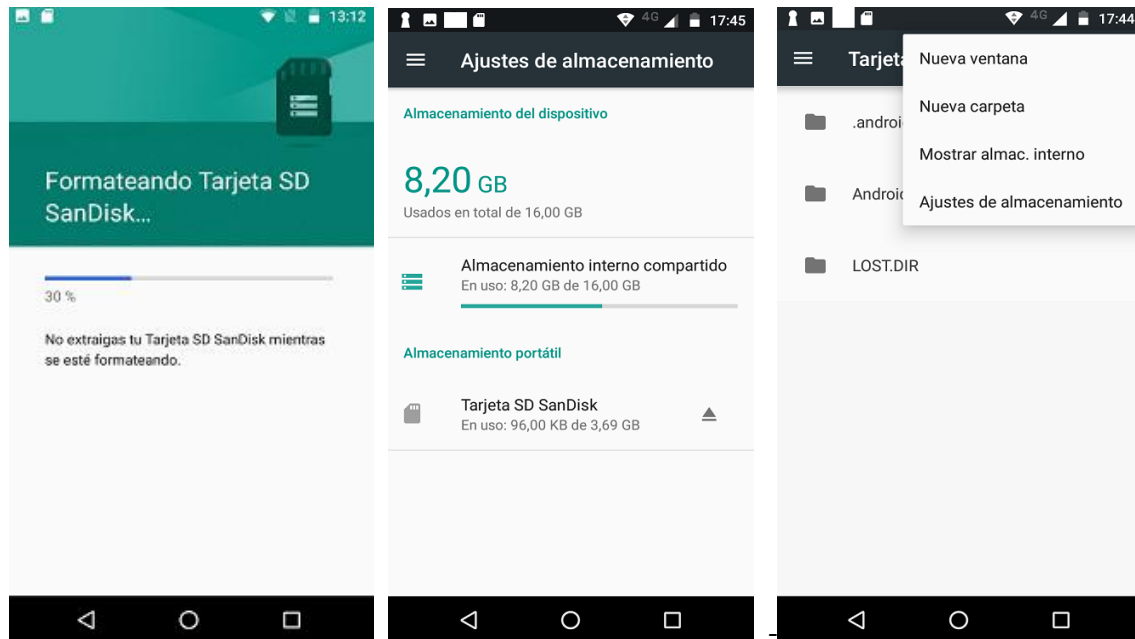
1080. Los dispositivos OTG sólo pueden configurarse como almacenamiento portátil, salvo que se fuerce lo contrario mediante un comando ADB específico (detallado posteriormente).
1081. El tipo de configuración de una tarjeta SD externa que ya ha sido definida en el dispositivo móvil puede cambiarse posteriormente a través de "Ajustes - [Dispositivo] Almacenamiento y USB". Esta operación requerirá formatear de nuevo la tarjeta, por lo que los datos existentes se perderán.

10.2.1. ALMACENAMIENTO ADOPTABLE EXTERNO

1082. Dado que los dispositivos Nexus como el empleado en la presente guía no tienen ranura física para la inserción de una tarjeta SD externa, la configuración e imágenes que se detallan a continuación corresponden a la configuración de un terminal de otro fabricante, por lo que pueden no coincidir exactamente con las pantallas que se muestren en un terminal de otros fabricantes de dispositivos móviles Android.
1083. El almacenamiento externo definido como portátil es uno de los mecanismos disponibles en Android para compartir datos y ficheros entre apps, almacenar contenidos multimedia por parte del usuario (como por ejemplo fotos y vídeos), y para compartir datos con un ordenador conectado por USB al dispositivo móvil.
1084. Al insertar una tarjeta SD externa en un dispositivo móvil que presente una ranura física, se lanzará una notificación similar a la mostrada a continuación. Una vez se seleccione la notificación, se lanzará el menú que permite configurar la tarjeta SD (por defecto, la opción para usar la tarjeta como almacenamiento portátil estará seleccionada):



1085. El formateo de una tarjeta SD externa que se va a usar como almacenamiento portátil en un dispositivo móvil que presente una ranura física se llevará a cabo usando FAT32, por lo que su contenido será accesible desde cualquier otro dispositivo que pueda montar un sistema de archivos FAT o VFAT.



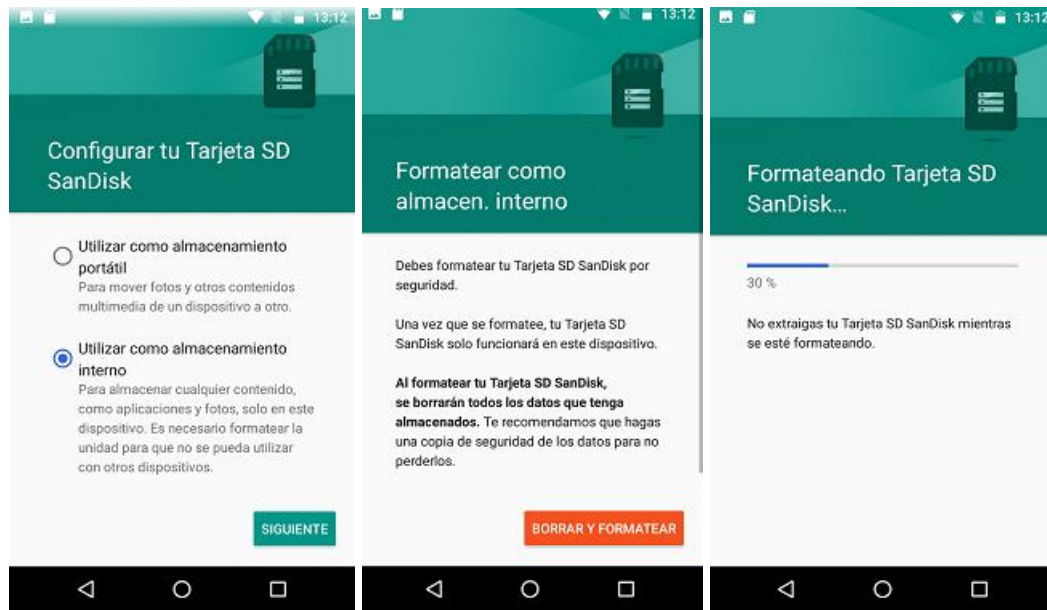
1086. Una vez configurada la tarjeta, es posible acceder a ella a través de "Ajustes - [Almacenamiento y USB] <Nombre de la tarjeta> - <...> - Ajustes de almacenamiento", como se ilustra en las imágenes previas.
1087. Adicionalmente, debe tenerse en cuenta que el sistema de ficheros (FAT o FAT32; VFAT) empleado por defecto en la tarjeta de almacenamiento externa configurada como portátil no establece permisos ni impone ninguna restricción, por lo que cualquier app con permisos de acceso sobre la tarjeta SD dispondrá de acceso a todos los datos almacenados en ella (de lectura o de escritura).
1088. En los dispositivos móviles que no disponen de capacidades de almacenamiento externas reales, en forma de un *slot* o ranura para tarjetas SD removibles, sin embargo, también se dispone de almacenamiento externo desde el punto de vista de Android, entendiendo como tal un espacio de almacenamiento virtual o emulado en la memoria interna (y, por tanto no removable) que se asocia a una partición anclada al punto de montaje `/sdcard` (o `/storage`). Esta partición se trata como si fuera una tarjeta de almacenamiento externa, por lo que se puede acceder a ella desde un ordenador que se conecte por USB al dispositivo móvil. Aunque este área de almacenamiento se trata como si fuera almacenamiento externo, realmente reside en la memoria interna del dispositivo móvil. En él se almacenan, entre otros, las fotografías y capturas de pantalla.
1089. Desde un equipo Windows, la referencia a `/sdcard` en el sistema de ficheros (que normalmente apunta a `/data/media/0` o a `/storage/sdcard0s`) se gestionará como una unidad de almacenamiento FAT32 al conectar el dispositivo móvil por USB a un ordenador, aunque en realidad la partición donde reside hace uso del sistema de ficheros ext4 (y se emplea FUSE para la conversión entre distintos sistemas de ficheros⁷⁷).

⁷⁷ <http://forum.xda-developers.com/google-nexus-5/general/info-storage-nexus-5-data-info-loss-t2534010>

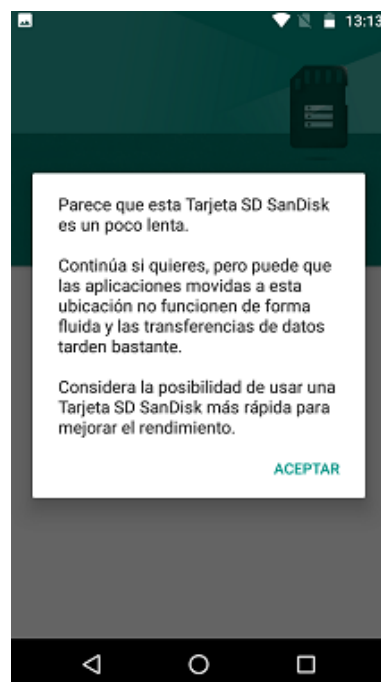
10.2.2. ALMACENAMIENTO ADOPTABLE INTERNO

1090. El almacenamiento adoptable (típicamente asociado a tarjetas SD que se insertan en una ranura física del dispositivo móvil) que se configure como interno tiene como misión expandir la memoria física del dispositivo, por lo que sobre él se podrán instalar aplicaciones y sus datos asociados.
1091. Es importante reseñar que el almacenamiento adoptable configurado como "interno" no debería extraerse del dispositivo, ya que se monta como parte del sistema de ficheros ext4 cuando se arranca el terminal, y dará lugar a fallos si deja de estar accesible.
1092. El almacenamiento adoptable interno también se marca como almacenamiento preferido, de forma que todos los datos que se generen en el dispositivo a partir de entonces pueden ser ubicados en él, por lo que extraerlo del terminal ocasionará fallos graves.
1093. Si la memoria física del dispositivo está cifrada, el contenido de un dispositivo configurado como almacenamiento interno también lo estará. En este caso, el dispositivo externo se cifrará mediante una clave AES de 128 bits, la cual se almacena en la memoria interna del dispositivo móvil, lo que impide que pueda montarse en otro dispositivo distinto a menos que se extraiga la clave del dispositivo original.
1094. Las apps que quieren leer o escribir en el almacenamiento adoptable interno deben solicitar los permisos "READ_EXTERNAL_STORAGE" y "WRITE_EXTERNAL_STORAGE" respectivamente ⁷⁸, siendo únicamente necesario el permiso de escritura para ambas tareas (ya que éste conlleva explícitamente el permiso de lectura).
1095. Desde Android 4.4 estos permisos no son necesarios si la app sólo requiere disponer de acceso a ficheros privados de la propia app, mediante el uso de directorios privados a la app dentro del almacenamiento externo.
1096. Desde Android 4.x se dispone (potencialmente) de capacidades para el cifrado de los datos que vayan a ser guardados en tarjetas de almacenamiento externas de forma nativa.
1097. Con el objetivo de evitar el acceso a los datos almacenados en tarjetas externas por parte de un potencial atacante, se recomienda habilitar esta opción de cifrado.
1098. Si, tras insertar una tarjeta SD en un dispositivo móvil, se opta por darla de alta como almacenamiento interno, el sistema Android solicitará su formateo (que implica el borrado de todos los datos que se encontrasen en la tarjeta) y preguntará al usuario si desea transferir datos al almacenamiento adoptado:

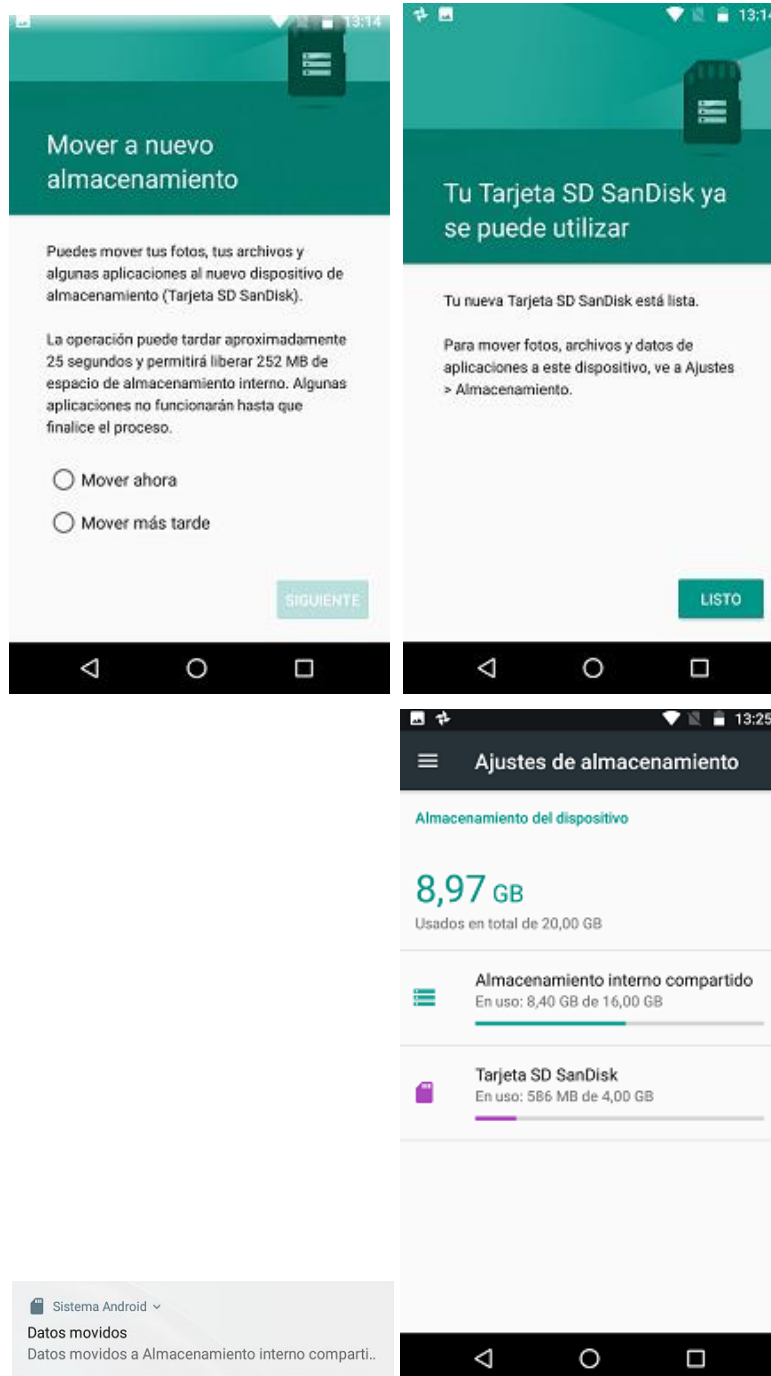
⁷⁸ <http://developer.android.com/guide/topics/data/data-storage.html#filesExternal>



1099. El sistema Android computará ciertas medidas de rendimiento y, si se detecta que la tarjeta es demasiado lenta, mostrará el siguiente aviso:



1100. Si se opta por mover parte del contenido del almacenamiento interno del dispositivo móvil a la tarjeta adoptada, se mostrará la siguiente notificación, y la nueva tarjeta se mostrará en el menú de "Ajustes - [Dispositivo] Almacenamiento - [Almacenamiento del dispositivo] <dispositivo>":

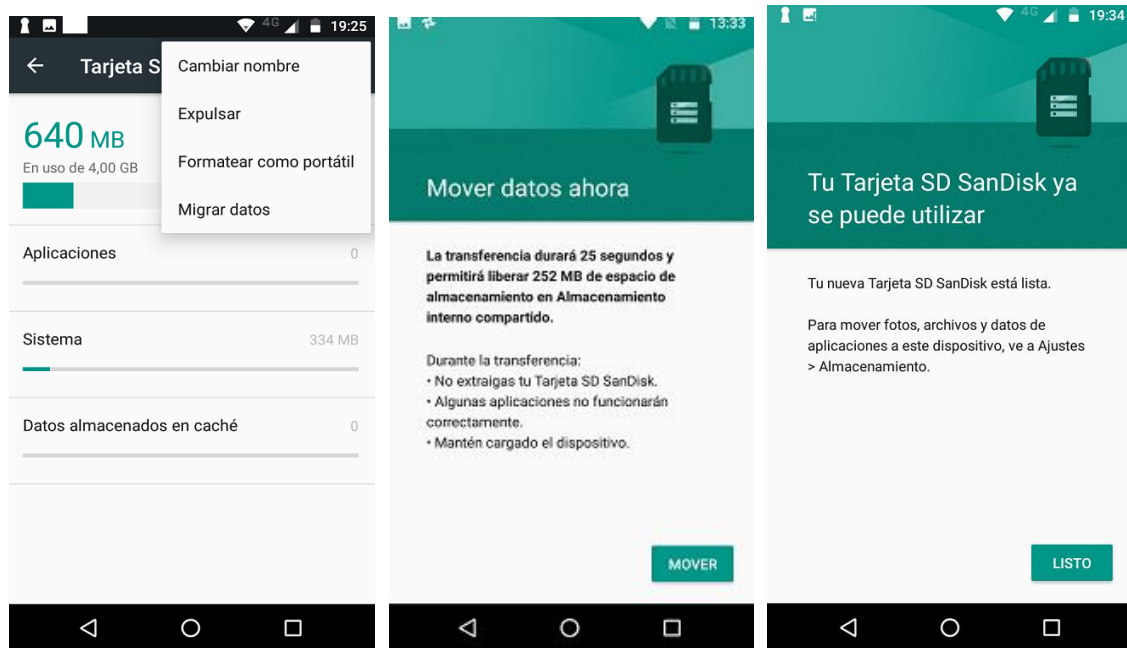


1101. Al igual que ocurría en versiones previas de Android (desde Android 2.2 o nivel de API 8) [Ref.- 612], de nuevo, las apps pueden ser movidas por el usuario manualmente desde la memoria interna a la memoria externa (configurada como interna), y viceversa.
1102. En el caso de Android 2.2 los ficheros binarios (DEX), los datos privados de la app, y las librerías nativas permanecen aún en el almacenamiento interno [Ref.- 613]. Por tanto, la principal diferencia es que en Android 7.x también los datos privados de la app pueden ser almacenados en el almacenamiento externo adoptado.
1103. Asimismo, en Android 2.2 se empleaban contenedores cifrados individuales para cada app, haciendo uso de una clave aleatoria generada y

asociada al dispositivo móvil, y por tanto, no permitiéndose el uso de la tarjeta SD en otro dispositivo.

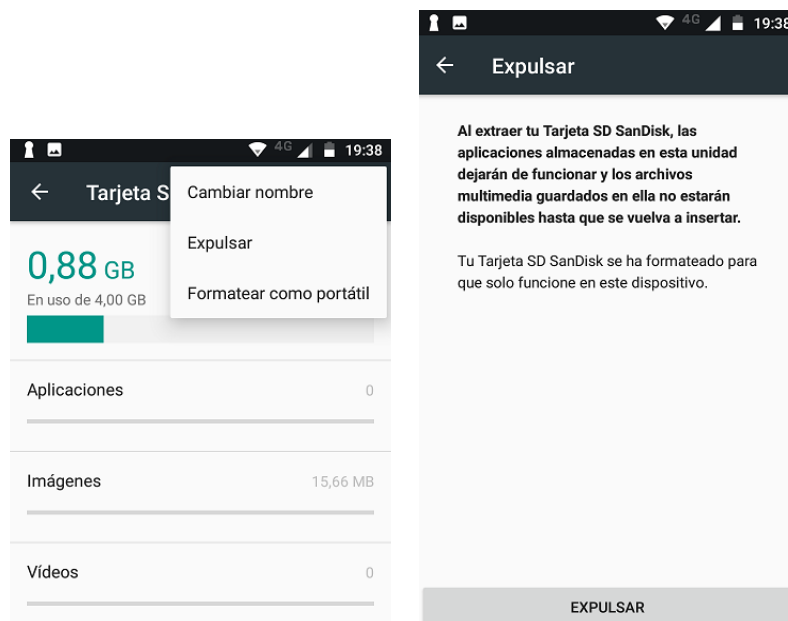
1104. Desde Android 6.x, sin embargo, se cifra el dispositivo de almacenamiento adoptado que se usa como almacenamiento interno al completo. El cifrado de la tarjeta SD (o de cualquier otro tipo de dispositivo de almacenamiento adoptable configurado como interno) no utiliza un cifrado basado en el TEE, a diferencia del cifrado del almacenamiento propio del dispositivo móvil. Por tanto, si la clave de cifrado se extrae del dispositivo móvil, los datos podrían accederse desde un equipo externo, existiendo la posibilidad de montarlo en un equipo o dispositivo diferente si se hace uso de la clave extraída asociada al almacenamiento adoptado [Ref.- 614].
1105. La clave estática empleada para el cifrado de todo el volumen o partición de datos del almacenamiento adoptado está disponible en el fichero `"/data/misc/vold/expand_<hash>.key"` (únicamente accesible por root). Esta clave corresponde a una clave de 128 bits empleada por dm-crypt al hacer uso del algoritmo AES 128 bits, "aes-cbc-essiv:sha256", de manera similar a la implementación de FDE.
1106. Android respetará la preferencia de la app establecida en su fichero *manifest*⁷⁹ para su instalación, respecto a la localización en las unidades de almacenamiento, no pudiendo instalarla en el almacenamiento externo si la app indica que sólo quiere ser instalada en el interno ("internalOnly").
1107. En Android 7.x, conviene recordar que los dispositivos de almacenamiento externos que se configuren como portátiles no se cifran. Por tanto, desde el punto de vista de seguridad, no se recomienda almacenar en ellos ningún contenido sensible.
1108. Se ha de tener cautela a la hora de extraer una tarjeta adoptable configurada como almacenamiento interno, ya que, si se extrae sin haberla desmontado correctamente, algunas aplicaciones dejarán de funcionar, e incluso se puede corromper la memoria interna del dispositivo móvil.
1109. Si se desea eliminar del terminal un dispositivo adoptable configurado como almacenamiento interno, se debe asegurar que los datos residentes en él se migran al almacenamiento interno del terminal mediante "Ajustes - [Dispositivo] Almacenamiento - <seleccionar el dispositivo adoptable> - <...> - Migrar datos":

⁷⁹ <https://developer.android.com/guide/topics/manifest/manifest-element.html#install>

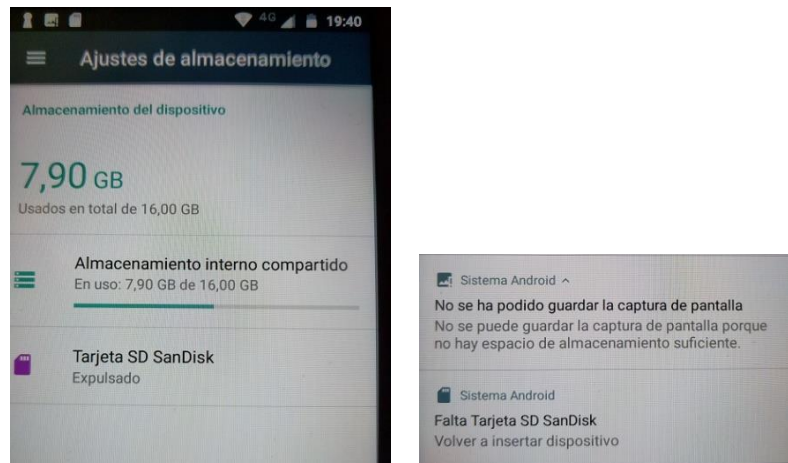


1110. Si, durante esa operación, alguna aplicación que acceda al dispositivo adoptable trata de acceder a él, fallará (como en caso de tomar una captura de pantalla) y el sistema mostrará una notificación informativa.

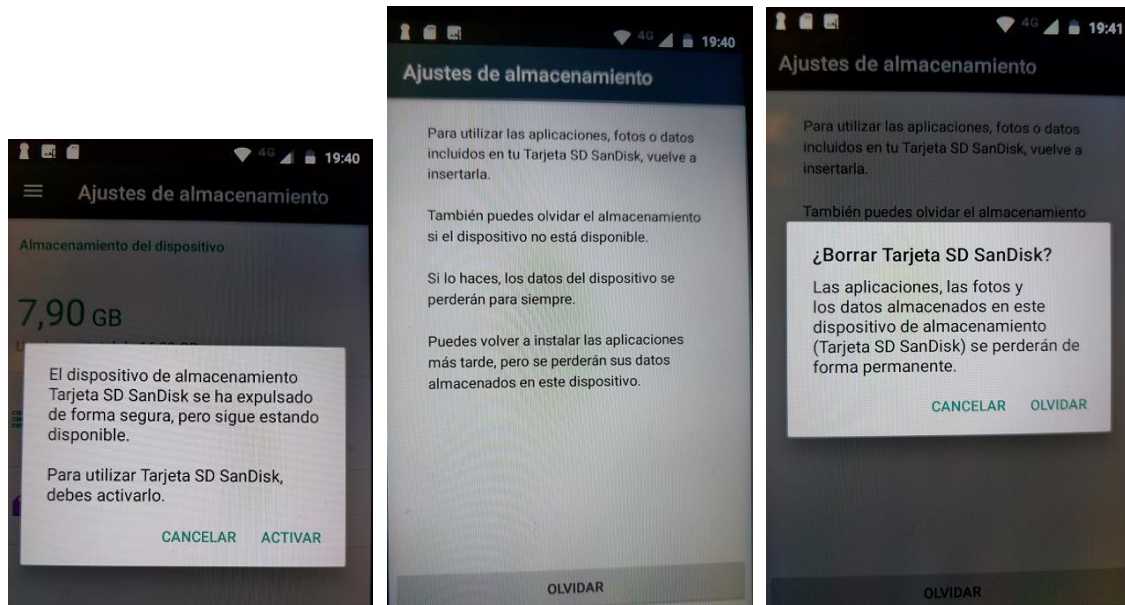
1111. Una vez migrados los datos al almacenamiento interno del dispositivo móvil, se podrá expulsar la tarjeta SD mediante "Ajustes - [Dispositivo] Almacenamiento - <seleccionar el dispositivo adoptable> - <...> - Expulsar":



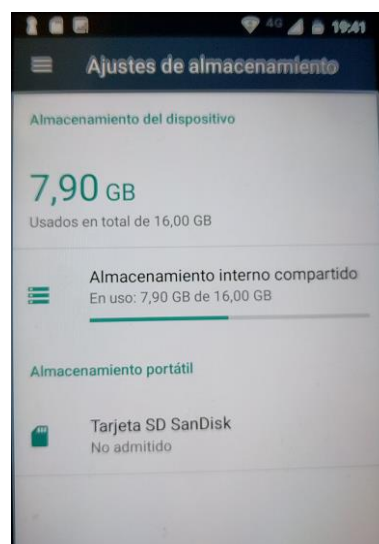
1112. Pese a que se expulse correctamente, el sistema Android podrá seguir mostrando errores y solicitar que se vuelva a insertar el dispositivo externo:



1113. Para resolver el problema, se debe abrir la notificación "Falta tarjeta SD <nombre>" y seleccionar el botón "Cancelar", lo cual despliega una ventana que permite "Olvidar" el dispositivo y confirmar la acción:



1114. Tras completarse esta acción, la tarjeta será vista dentro del apartado "Almacenamiento portátil" como "No admitido":



1115. Teniendo en cuenta todas las consideraciones anteriores, se recomienda utilizar con cautela el almacenamiento adoptable interno, pues puede dar lugar a una corrupción del sistema de ficheros que solo se podría resolver restableciendo el terminal a fábrica.

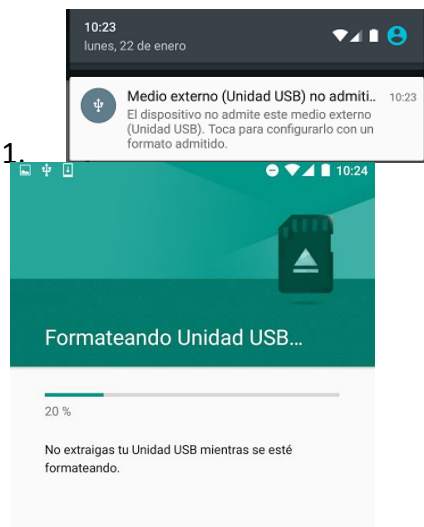
10.3. DISPOSITIVOS DE ALMACENAMIENTO OTG (ON-THE-GO)

1116. Numerosos dispositivos móviles comercializados con Android no disponen de ranura física que permita la inserción de una tarjeta SD que pueda ser empleada como almacenamiento complementario al del dispositivo, como por ejemplo el dispositivo Nexus empleado en la elaboración de la presente guía.
1117. USB OTG (On-the-Go) es una norma que extiende el protocolo USB 2.0 para invertir el papel del puerto USB, de forma que el dispositivo móvil pase de tener rol de esclavo al rol de *host* en la conexión. Ello permite que el dispositivo móvil pueda acceder como maestro a otros dispositivos, como un disco USB, un *pendrive*, etc.
1118. Por defecto, los dispositivos conectados a través de USB OTG no pueden ser configurados como almacenamiento interno. Si se desea utilizarlos de este modo, se necesita ejecutar el siguiente comando mediante ADB [Ref.-614]:

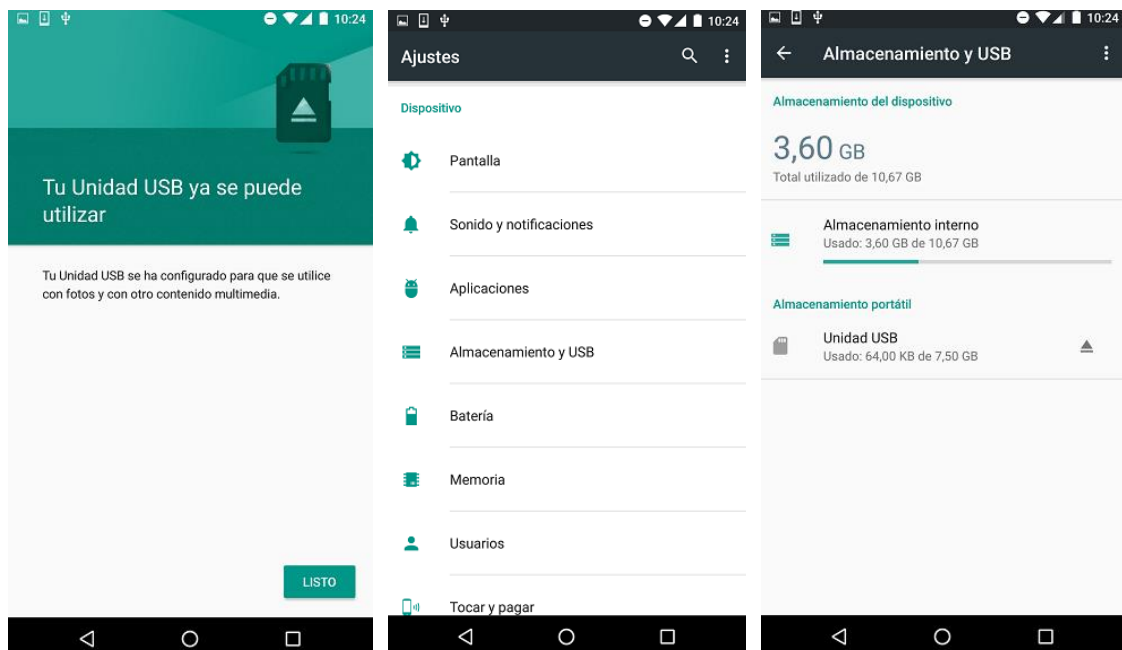
```
C:\> adb shell sm set-force-adoptable true
```

1119. No obstante, el uso de dispositivos OTG como almacenamiento interno no es práctico ni recomendable, ya que una expulsión fortuita del dispositivo provocaría un fallo del sistema y, potencialmente, la corrupción del sistema de ficheros que forma parte de la memoria interna. Se recomienda pues utilizar los dispositivos OTG como almacenamiento portátil.
1120. Para configurar un dispositivo OTG como almacenamiento portátil, Android 7.x no requiere habilitar ninguna opción oculta. Al conectar un dispositivo OTG al terminal móvil, se mostrará una notificación que permite iniciar su configuración:

1121.

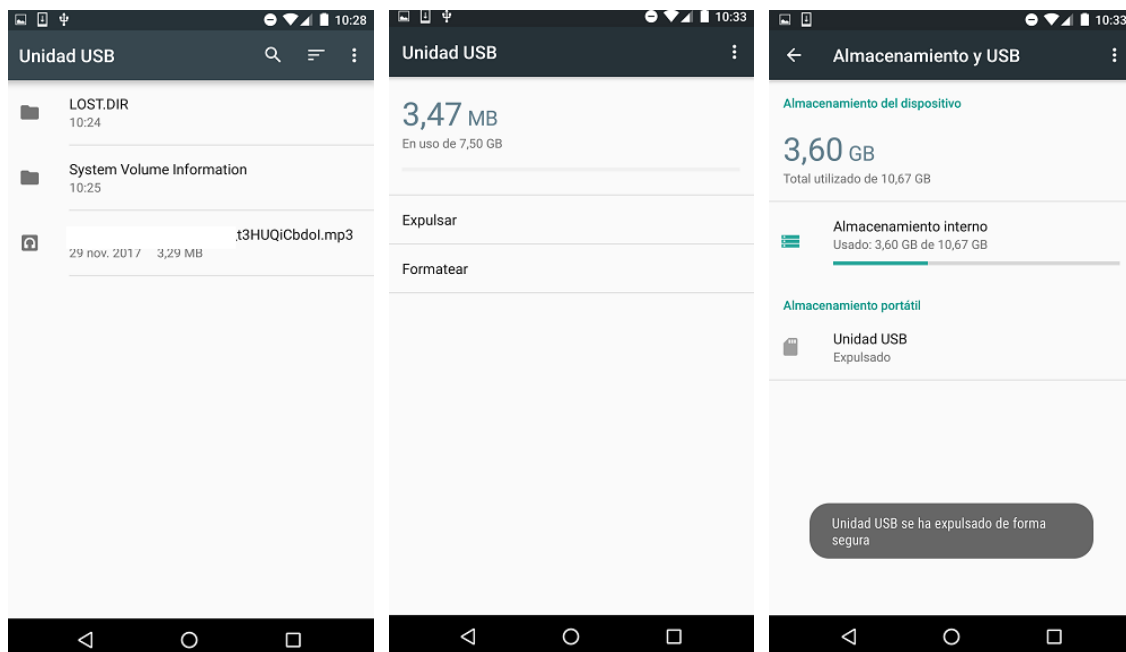


1122. Una vez concluya el formateo, la unidad estará accesible desde "Ajustes - [Dispositivo] Almacenamiento - [Almacenamiento portátil] Unidad USB":



1123. El botón situado a la derecha del nombre del almacenamiento portátil ("Unidad USB" en el ejemplo) permite expulsar el dispositivo externo de

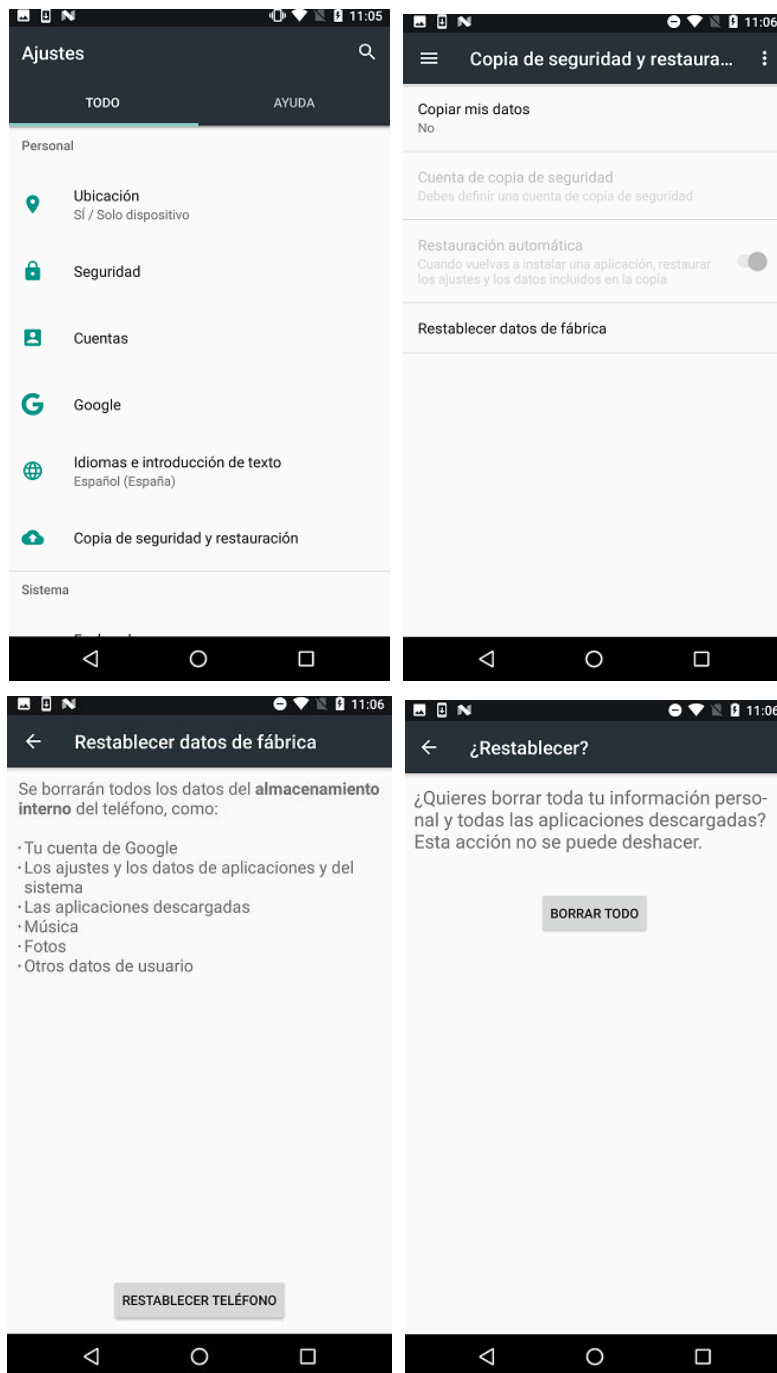
forma segura. Si se pulsa sobre el nombre del dispositivo externo, se abrirá un explorador que permitirá ver sus contenidos, junto al menú "... " que da acceso a su formateo y a su expulsión:



1124. En Android 7.x no existe ninguna opción para cifrar el contenido de un dispositivo de almacenamiento portátil, por lo que, desde el punto de vista de seguridad, no se recomienda utilizarlos para guardar información potencialmente sensible, sino únicamente contenidos que puedan llevarse a cualquier otro dispositivo sin comprometer la privacidad ni la seguridad de su propietario.

10.4. ELIMINACIÓN DE DATOS DEL DISPOSITIVO MÓVIL

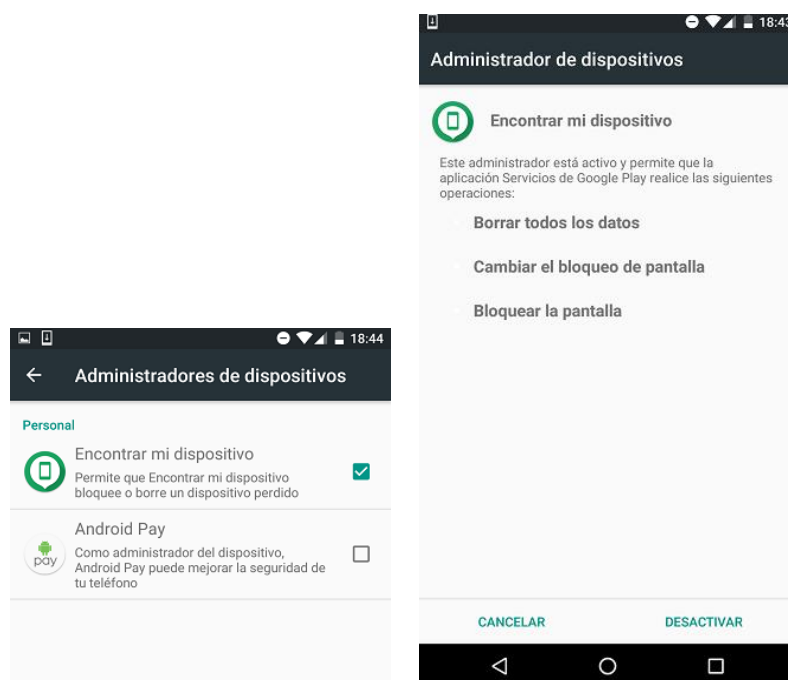
1125. El objetivo de las capacidades de cifrado de Android es evitar que un tercero sin autorización disponga de acceso a los datos almacenados en el dispositivo móvil. Estas capacidades pueden ser complementadas con la funcionalidad que permite, tanto de forma local como remota, eliminar los datos del usuario del dispositivo móvil.
1126. Android 7.x proporciona capacidades locales para la eliminación de todos los datos almacenados en el dispositivo móvil a través del menú "Ajustes [Personal] - Copia de seguridad" y la opción "Restablecer datos de fábrica". Antes de poder lanzar la operación, el dispositivo móvil solicitará el código de acceso:



1127. El proceso de restablecimiento eliminará todos los datos del teléfono, incluyendo cuentas de Google, configuración y datos de apps y del sistema, así como apps de terceros previamente descargadas e instaladas. No se eliminará el software del sistema operativo ni las actualizaciones de sistema ya aplicadas, es decir, el propio Android, ni los datos de las tarjetas de almacenamiento externas. También se eliminarán datos de otros usuarios si se han empleado las capacidades multi-usuario descritas en el apartado "9.19. Múltiples perfiles de usuario".
1128. Para confirmar esta operación crítica es necesario introducir el código de acceso tras seleccionar el botón "Borrar Todo" de la pantalla que informa de que se borrarán todos los datos. Si existen otros usuarios en el dispositivo

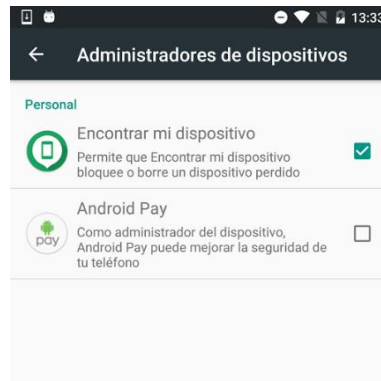
móvil, también se informará de este hecho por si se desea informarles antes de proceder al borrado de los datos.

1129. Es importante reseñar que, si no existe un código de acceso establecido en el dispositivo móvil, la operación continuará tras pulsar el botón "Borrar Todo". Por tanto, una vez más, se aconseja vehementemente disponer de un código de acceso al terminal.
1130. Estas capacidades de borrado de datos local también pueden ser activadas tras llevarse a cabo un número determinado de intentos de acceso incorrectos, pudiendo configurarse a través de las políticas de seguridad de los mecanismos de gestión empresarial de dispositivos móviles Android (ver apartado "8. Gestión empresarial de dispositivos móviles basados en Android").
1131. Las aplicaciones administradoras del dispositivo móvil pueden ser activadas o desactivadas en función de las necesidades del usuario. Antes de su ejecución, siempre verificarán si están activadas, y en caso de no estarlo, solicitarán al usuario su activación. Para verificar el estado de dichas aplicaciones, se puede acceder al menú "Ajustes - [Personal] Seguridad - [Administración de dispositivos] Administradores de dispositivos":



1132. Además de mediante las capacidades de gestión empresarial de dispositivos móviles, el propietario de un dispositivo que lo haya asociado a su cuenta de Google puede efectuar labores de gestión remota (incluyendo el borrado de datos del dispositivo) siguiendo las indicaciones descritas en el apartado "8.1. Gestión remota de dispositivos Android por parte del usuario".
1133. En Android 6.x, se incorporó "Android Pay" (que pasó a denominarse "Google Pay" posteriormente) como un administrador de dispositivos. El objeto es permitir que los datos almacenados como medio de pago en

Android Pay puedan ser borrados de forma remota en caso de pérdida o robo del dispositivo móvil, protegiendo así al usuario:



10.5. DIRECT BOOT

1134. El modelo de arranque Direct Boot surge a la par que el cifrado FBE (ver apartado "10. Cifrado de datos en Android"), para permitir que un dispositivo móvil pueda ofrecer parte de su funcionalidad sin que el usuario haya introducido el código de acceso, empleado para el cifrado, durante el proceso de arranque.

1135. Con anterioridad a Android 7.x, cuando se cifraba el dispositivo móvil (empleando el cifrado FDE introducido por Android 3.0 y establecido como obligatorio a partir de Android 6.0 para los dispositivos que incorporasen esta versión de fábrica), tras un re arranque no iniciado por el usuario, las apps no disponían de acceso a la partición de datos ("/data"), por no poder ésta ser descifrada y montada hasta que el usuario introdujese el código de acceso. Esto implica que no se recibirán notificaciones, alarmas, llamadas, etc., hasta que el usuario es consciente de la situación e introduce el código de acceso, se descifra la partición de datos, y se completa el proceso de arranque.

1136. Otro inconveniente de dicho modelo afecta a los servicios de accesibilidad, que no están disponibles hasta que se pueda acceder a la partición de datos.

1137. Como medio para evitar estos inconvenientes al hacer uso de FDE, se permite deshabilitar la opción "Inicio seguro" durante la configuración del código de acceso mediante PIN, patrón o contraseña. Si se selecciona la opción "No, gracias", Android utilizará la clave "default_password" (idéntica para todos los dispositivos Android) como contraseña para proteger la clave maestra, utilizada para descifrar y montar la partición de datos ("/data") y permitir la recepción de notificaciones, alarmas y llamadas. Con este modelo, una vez iniciado el sistema operativo Android, para poder acceder al dispositivo móvil será necesario introducir el código de acceso en la pantalla de bloqueo.

1138. Por otro lado, si se habilitan los servicios de accesibilidad, el dispositivo móvil se saltará el ajuste de inicio seguro "Requerir <código> para iniciar

dispositivo", y el terminal presentará directamente la pantalla de bloqueo (de manera equivalente a la opción "No, gracias" previamente descrita).

1139. Con el modelo FBE, en lugar de requerirse que la partición de datos se descifre de forma global, se permite que el sistema pueda descifrar en el arranque sólo los ficheros que son necesarios, y no toda la partición. Las apps esenciales, que ofrecen capacidades básicas (alarmas, notificaciones y recepción de llamadas), pueden permanecer ejecutando con funcionalidad limitada tras el arranque del dispositivo móvil, y una vez el usuario introduce el código de acceso, pasar a disponer de toda su funcionalidad al poder acceder a todos los datos del usuario.
1140. Para que una app que no es de sistema pueda realizar una acción en modo Direct Boot es preciso que registre los componentes que deben ejecutar en ese modo como *"encryption aware"* asignando el valor *"true"* (verdadero) al atributo *android:directBootAware* de su fichero manifest [Ref.- 735].
1141. Para las apps de sistema, el atributo disponible es *"defaultToDeviceProtectedStorage"*. Al asignarle valor verdadero, el sistema asignará el DE como almacenamiento por defecto de la app, en lugar del CE (ver apartado "10. Cifrado de datos en Android"). Desde el punto de vista de seguridad, es crucial que las apps de sistema únicamente almacenen los datos no sensibles en el DE (Device Encrypted), y sigan recurriendo al CE (Credential Encrypted) para los datos que deban permanecer privados para la app (y en ningún caso, dejar que se almacene información personal en el DE).
1142. En caso de requerir acceder a datos, la app deberá utilizar el almacenamiento cifrado de tipo DE, el cual solo está disponible si el dispositivo ha realizado un arranque seguro (ver apartado "7.9. Proceso de arranque seguro: Verified Boot").
1143. Los componentes de una app designados como *"encryption aware"* pueden registrarse para recibir del sistema el mensaje de tipo broadcast que informa de que el dispositivo ha completado el re arranque.
1144. Es importante añadir que el acceso a datos en modo Direct Boot es restringido: así por ejemplo, aunque se permite recibir llamadas, no se mostrará la información del contacto que la realiza; en el caso de la recepción de SMS, se recibirá la notificación de su llegada, pero no se mostrarán los detalles hasta que el dispositivo no sea desbloqueado.
1145. Una vez que el usuario desbloquea el dispositivo tras un reinicio, la app que haya registrado algún componente para recibir la notificación del desbloqueo y poder acceder al almacenamiento cifrado de tipo CE.
1146. De cara a los desarrolladores de apps que hacen uso de Direct Boot, la recomendación es que los datos que almacenen en el almacenamiento de tipo DE no sean sensibles, ya que estarán disponibles para cualquier app que ejecute en dicho modo.

11. GESTIÓN DE CERTIFICADOS DIGITALES Y CREDENCIALES

11.1. ANDROID KEYSTORE SYSTEM: ALMACENAMIENTO DE CREDENCIALES

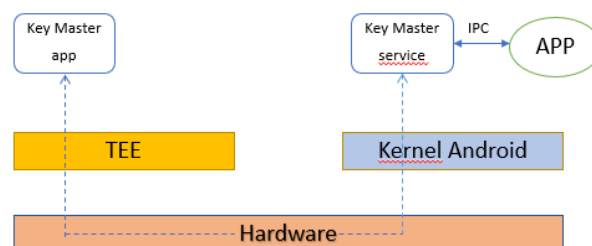
1147. Desde Android 1.6, Android dispuso de un almacén de credenciales a nivel de sistema. Este almacén sólo se usaba inicialmente para almacenar claves privadas y certificados Wi-Fi y VPN, y no existía ninguna API pública para acceder a él⁸⁰.
1148. Android 4.0 integró el almacén de credenciales en el sistema operativo, e introdujo una API pública que se denominó "KeyChain" que proporcionaba acceso a claves privadas y certificados gestionados por el sistema operativo. Esta API no proporcionaba mecanismos de gestión para las apps que requiriesen almacenar material criptográfico (claves, tokens, certificados, etc.).
1149. El acceso al almacén de credenciales se realizaba a través del demonio *keystore*, que cifraba las claves mediante una clave maestra AES 128 bits. Esta clave maestra se derivaba del mecanismo de control de acceso configurado en el dispositivo móvil (contraseña o PIN).
1150. Cada clave se almacenaba como un fichero residente en el directorio "/data/misc/keystore/". Las apps podían acceder a sus claves, pero no a las de otras apps (lo cual se garantizaba porque la clave se ligaba al UID del proceso que la creó).
1151. Al no estar la clave maestra ligada al dispositivo móvil, los ataques de fuerza bruta sobre los ficheros de claves copiados a otros dispositivos eran posibles.
1152. Oficialmente, este repositorio no proporcionaba funciones para el almacenamiento de contraseñas, u otras credenciales y secretos por parte de las apps⁸¹, o de claves simétricas hasta Android 6.x (y versiones posteriores) [Ref.- 621].
1153. Con la versión 4.3 de Android, el sistema de almacenamiento de credenciales (entendiendo como tal certificados digitales de CAs y, en todo caso, el par de claves pública y privada asociadas a los certificados digitales de usuario, personales o cliente) pasó a denominarse oficialmente "Android Key Store" [Ref.- 144].
1154. Para acceder al "Android Key Store", se publicó un interfaz (o API) que permitía a las apps almacenar certificados digitales, incluso independientes por usuario (en dispositivos Android multi-usuario). Esta API, asociada al HAL (Hardware Abstraction Layer), proporcionaba capacidades para firma digital, además de generación e importación de claves asimétricas, y permitía determinar si el almacenamiento de credenciales se gestionaba a nivel de

⁸⁰ <https://nelenkov.blogspot.com.es/2012/05/storing-application-secrets-in-androids.html>

⁸¹ <https://github.com/nelenkov/android-keystore>

software o en modo "*hardware-backed*", a criterio del fabricante y modelo del dispositivo móvil.

1155. El modo "*hardware-backed*" es aquel en el que el almacenamiento y gestión de las claves utilizadas por las apps se realiza en un almacenamiento hardware seguro dentro del dispositivo, de forma que dichas claves no pueden ser extraídas del mismo, ni siquiera mediante el kernel o disponiendo de acceso como root.
1156. La implementación del almacén seguro mediante hardware, por ejemplo en algunos dispositivos móviles Nexus basados en los SoC (System on a Chip) Qualcomm, emplea las capacidades TrustZone de los procesadores ARM y el TEE (Trusted Execution Environment), que en concreto en Qualcomm se denomina el Qualcomm Secure Execution Environment (QSEE) [Ref.- 191] [Ref.- 144][Ref.- 145]. Otras implementaciones son posibles haciendo uso de Secure Elements (SEs), Trusted Platform Modules (TPMs) o Universal Integrated Circuit Cards (UICCs; por ejemplo, una tarjeta SIM).
1157. En los dispositivos basados en Qualcomm, la app se comunica con el servicio Keymaster, quien a través del kernel, a su vez, gestiona el acceso a las claves con la app Keymaster existente en el TEE (este tipo de apps que ejecutan en el TEE se denominan habitualmente *trustlets*), responsable de las operaciones asociadas a dichas claves dentro del propio hardware, a través del TEE.



1158. Si la implementación del almacén es de tipo software ("*software-backed*"), las claves son cifradas mediante AES [Ref.- 189] con una clave maestra para cada usuario, y almacenadas bajo `"/data/misc/keystore/user_N/"` (donde por defecto "N" es igual a "0" para el usuario inicial o propietario del dispositivo móvil, es decir, `".../user_0/"`). En el siguiente ejemplo el nombre de los ficheros corresponde al nombre asignado al certificado, "user (CNI)":

```

root@mobile:/data/misc/keystore/user_0 # ls -l
-rw-----  keystore  keystore          1300   2018-06-20   21:07
1000_CACERT_user+P+XCNI+Y
-rw-----  keystore  keystore          1412   2018-06-20   21:07
1000_USRCERT_user+P+XCNI+Y
-rw-----  keystore  keystore          1652   2018-06-20   21:07
1000_USRPKEY_user+P+XCNI+Y
  
```

Nota: El dispositivo móvil utilizado en la elaboración de la presente guía no permite almacenamiento de credenciales por software, por lo que no es posible detallar la

estructura de ficheros asociados a las claves. La estructura final puede depender de la implementación del almacén de credenciales que realice cada fabricante.

Adicionalmente, se debe tener en cuenta que los requisitos de compatibilidad impuestos por Google requieren que los dispositivos móviles desde al menos la versión Android 7.x dispongan de soporte hardware para la gestión de claves criptográficas.

1159. Con Android 6.x (Marshmallow), se produjo una evolución importante del denominado "Android Key Store", que se convirtió en el "*Android Keystore System*" (en adelante, AKS), que es el subsistema que gestiona el material criptográfico del dispositivo móvil, tanto a nivel de sistema como de aplicaciones.

1160. El AKS permite almacenar material criptográfico en un contenedor o almacén de credenciales denominado a su vez *keystore*, protegido por mecanismos de cifrado. En el Keystore se almacenan, por ejemplo, los certificados digitales asociados a redes VPN y redes Wi-Fi (ver los apartados correspondientes en la presente guía), o los empleados desde el navegador web, el cliente de correo electrónico y otras apps. También permite almacenar claves secretas (mediante cifrado simétrico).

1161. Una app puede solicitar el uso de credenciales a través de:

- La API KeyChain [Ref.- 250], para el acceso a credenciales a nivel de sistema. A través de un interfaz presentado por el sistema, el usuario puede elegir a qué credencial puede acceder una app. La KeyChain, a su vez, hace uso del Keystore.
- El Android Keystore Provider (proveedor de Android Keystore, introducido con Android 4.3, API nivel 18), para el acceso individual a las credenciales. La app dispondrá de un Keystore propio para gestionar y almacenar sus credenciales, que sólo serán accesibles por ella misma [Ref.- 630]. No se requiere intervención del usuario para seleccionar la credencial a utilizar.

1162. Android 6.x extendió sus APIs para proporcionar nuevas capacidades al Keystore, como el uso de AES y HMAC, y un sistema de control de acceso para claves gestionadas por un dispositivo criptográfico de tipo hardware [Ref.- 621].

1163. Adicionalmente, se incorporaron parámetros para el control de acceso en la generación e importación de claves (tanto simétricas como asimétricas), los cuales se almacenan junto a las mismas como etiquetas (*tags*). Los controles de acceso están vigentes mientras la clave sea considerada válida de forma que, una vez generada o importada, no es posible cambiar el tipo de autorización. El Gatekeeper de Android permite definir el tipo de uso de las claves, que se puede englobar en las siguientes categorías:

- Criptografía: algoritmo de la clave y propósito de la misma (firma/verificación, cifrado/descifrado, etc.), el modo de cifrado de bloque y el tipo de *padding* (relleno) [Ref.- 629]. Si el dispositivo dispone de un TEE, estos elementos se controlarán a través de él. El Keystore soporta los tipos de claves definidos en KeyPairGenerator y KeyGenerator [Ref.- 745].

- Validez: período durante el cual se permite utilizar la clave, y es específico de cada clave. Dado que el TEE no dispone de un reloj independiente, las autorizaciones relacionadas con la validez de la clave no se realizan a través del TEE.
 - Autenticación: período durante el cual se puede usar la clave desde que el usuario introdujo las credenciales del dispositivo móvil (código de acceso) asociadas a ella. Se gestiona a través del TEE. Tras la autenticación del usuario, la clave puede estar configurada para:
 - Permitir su uso durante un período determinado: estas claves sólo están habilitadas si existe un bloqueo seguro configurado en el dispositivo. La clave estará disponible desde que el usuario valide el código de acceso vinculado a dicha clave y hasta que venza el plazo definido mediante `setUserAuthenticationValidityDurationSeconds`. Si se elimina el bloqueo seguro, estas claves quedarán inhabilitadas permanentemente.
 - Autorizar una determinada operación criptográfica asociada a la clave: sólo está disponible a través de identificación con huella dactilar, y cada operación específica requerirá autorización individual por parte del usuario. Si se añade una nueva huella, este tipo de clave quedará inhabilitada.
1164. Adicionalmente, se puede restringir el acceso a la clave bajo ciertas condiciones como, por ejemplo, el uso de dispositivos asociados a un sensor corporal⁸². Estas condiciones normalmente no se controlan por parte del TEE, sino por software.
1165. El AKS está diseñado de forma que las claves no puedan ser extraídas del dispositivo móvil, ni siquiera por las apps que las utilizan. Para ello, se restringe el material criptográfico a través de dos medidas de seguridad [Ref.- 630]:
- Las operaciones criptográficas que pueda requerir una app se llevan a cabo por un proceso de sistema, es decir, el material criptográfico se mantiene fuera del sandbox de las apps, y nunca se accede directamente por parte de sus procesos. De este modo, si el proceso de la app ha sido comprometido, el atacante sólo podrá usar las claves de esta app, pero en ningún caso extraerlas fuera del dispositivo.
 - El material criptográfico puede residir en un hardware específico si el dispositivo lo incorpora (por ejemplo, el TEE), de forma que nunca esté expuesto fuera de él; así, incluso aunque el sistema operativo Android se viese comprometido, un potencial atacante únicamente podría utilizar las claves del Keystore de cualquier app, pero no extraerlas del dispositivo para realizar ataques *off-device*.
1166. Las opciones existentes oficialmente para el almacenamiento de otros datos, secretos y contraseñas (aunque desaconsejado) por parte de las apps

⁸² <https://source.android.com/security/keystore/tags>

pasan por hacer uso de las preferencias compartidas (*SharedPreferences*) de la app, o del *AccountManager*⁸³ (gestor de cuentas), siempre debiendo ser muy cuidadosos los desarrolladores de apps de Android a la hora de almacenar información sensible en el *sandbox* de la propia app⁸⁴.

1167. En Android 6.x, el Keystore hace uso del Gatekeeper [Ref.- 631], un subsistema encargado de la gestión y verificación de los códigos de acceso y de tokens de autorización en el TEE. Tras la introducción de una clave por parte del usuario, el subsistema Gatekeeper, a través de la clave secreta derivada del TEE, firma una declaración de autenticación que permite a una app acceder a las claves que ha almacenado. A nivel de sistema, existe un demonio "*gatekeeperd*" que ejecuta como proceso independiente y que proporciona a las APIs de Android acceso al HAL y al Keystore.
1168. La versión 1 del Keymaster de Android no ofrecía ningún mecanismo para que las apps pudieran saber si el almacenamiento de las claves se realizaba en modo *hardware-backed*. El demonio del Keystore confiaba en el HAL en lo relativo al respaldo hardware de las claves [Ref.- 621].
1169. En Android 7.x, la versión 2 del Keymaster introdujo la denominada "atestación de claves" (*key attestation*), que proporciona un mecanismo para saber si una clave asimétrica se gestiona en modo *hardware-backed*, así como para conocer sus propiedades y sus restricciones de uso [Ref.- 621].
1170. Para ello se utiliza el denominado "certificado de atestación" (*attestation certificate*), que se firma con una clave de atestación que viene de fábrica y que opcionalmente contiene una referencia a la clave para la cual se emite el certificado, denominada "attestation extension" (extensión de atestación).
1171. Una extensión de atestación incluye las autorizaciones asignadas a la clave en el Keymaster, identificadas mediante una serie de etiquetas y propiedades. Entre ellas, se encuentra el atributo "teeEnforced", que permite saber cuáles proceden del TEE.

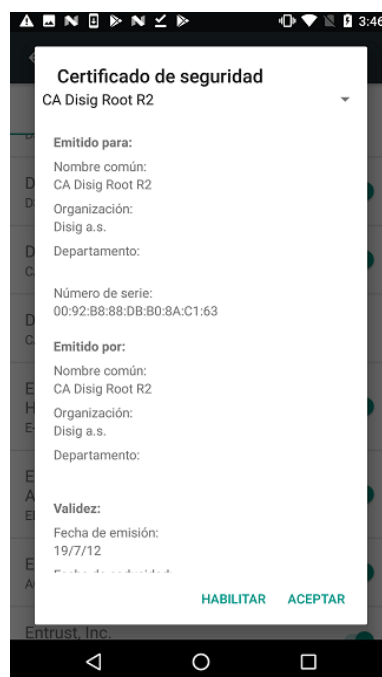
11.2. CERTIFICADOS DIGITALES DE SISTEMA Y DE USUARIO

1172. Un certificado digital es un mecanismo electrónico que acredita la identidad de su titular ante cualquier otra entidad que lo solicite. Para ello, el certificado digital debe venir firmado electrónicamente por una autoridad certificadora (CA), la cual emite el par de claves pública y privada asociadas al certificado.
1173. Un certificado digital puede ser de usuario o de sistema. En Android, los certificados de sistema corresponden a la lista de autoridades certificadoras (CAs) raíz reconocidas por defecto en esa versión concreta de Android. Los certificados de usuario corresponden a los certificados digitales tanto de CAs como personales que han sido añadidos posteriormente por el usuario.

⁸³ <http://developer.android.com/reference/android/accounts/AccountManager.html>

⁸⁴ <http://android-developers.blogspot.com.es/2013/02/using-cryptography-to-store-credentials.html>

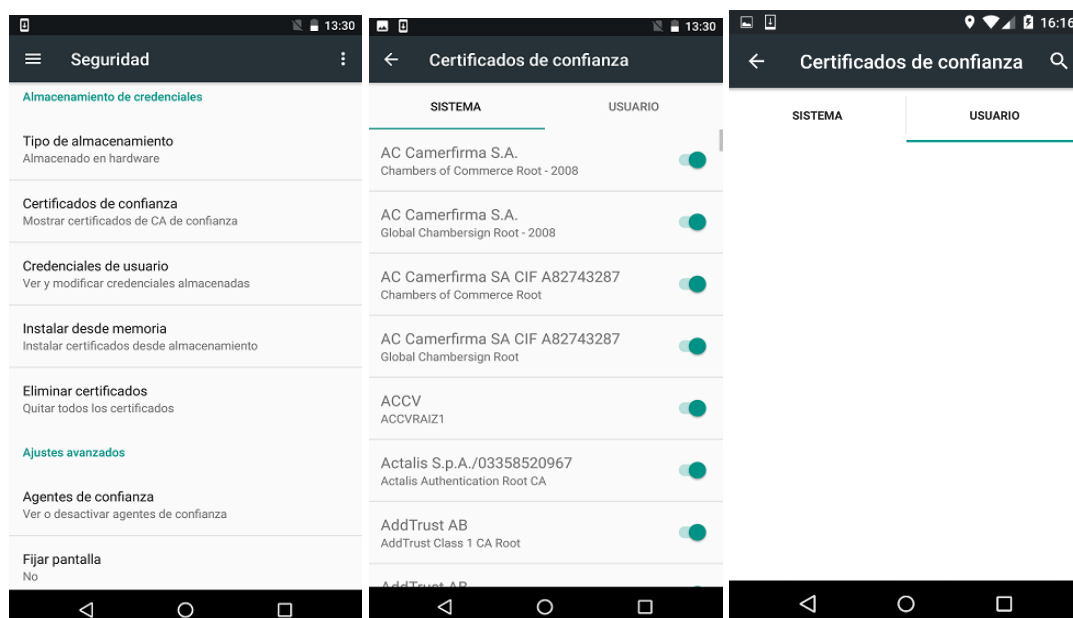
1174. Los certificados digitales se almacenan en el repositorio de credenciales.
1175. Android 7.x gestiona de manera más estricta las autoridades certificadoras (CAs) de confianza, de forma que las apps basadas en la API 24 ya no confiarán por defecto en las CAs añadidas por el usuario o el administrador (tal como sucedía hasta Android 6.0.1), sino que solo confiarán por defecto en las CAs de sistema. Una app confiará en las CAs añadidas por el usuario si lo solicita explícitamente, reduciendo así la superficie de ataque frente a ataques de MitM.
1176. Android 7.x introduce la restricción de que los dispositivos compatibles según los requisitos de Google sólo confíen en las CAs de sistema proporcionadas en el AOSP. Con esta medida, se evita que las CAs preinstaladas varíen de unos dispositivos, y/o fabricantes, a otros.
1177. Si una app necesita acceder a otras CAs no incluidas en el AOSP, puede hacerlo incluyendo sus CAs de confianza (denominadas "anclajes de confianza" o trust-anchors) en su fichero "network-security-config.xml"; en la sección "<network-security-config>" la app declarará en qué CAs confiará para las conexiones hacia un dominio específico, y podrá establecer limitaciones en las relaciones de confianza hacia ciertos dominios.
1178. A nivel de usuario, la sección "Almacenamiento de Credenciales" está disponible desde el menú "Ajustes - [Personal] Seguridad", en la parte inferior:



1179. La opción "Tipo de almacenamiento" indica los detalles de implementación del repositorio de credenciales. En el caso del dispositivo móvil bajo estudio, las credenciales se almacenan en un almacén seguro

implementado en el hardware del dispositivo: "Almacenado en hardware" (*hardware-backed*)⁸⁵.

1180. Los certificados de sistema pueden ser visualizados a través del menú "Ajustes - [Personal] Seguridad - [Almacenamiento de Credenciales] - Certificados de confianza".
1181. Hasta Android 6.x inclusive, los certificados de usuario (también denominados personales o certificados cliente) instalados por el propio usuario sólo se podían visualizar a través de los menús de las apps que hacen uso de ellos (por ejemplo, la configuración de redes VPN (ver apartado "19.6. Redes VPN") o de redes Wi-Fi basadas en 802.1x EAP (ver apartado "16.3. Recomendaciones de seguridad para la conexión a redes Wi-Fi").
1182. En Android 7.x, los certificados instalados por el usuario, tanto personales como de CAs, se pueden visualizar en la sección "Credenciales de usuario" bajo el menú "Ajustes [Personal] - Seguridad [Almacenamiento de Credenciales]".



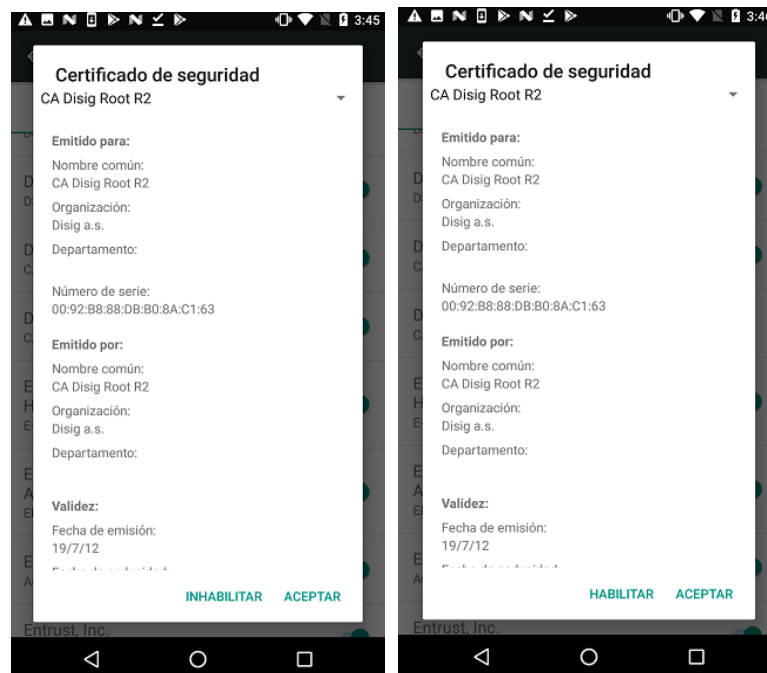
1183. En las versiones 2.x de Android, por defecto, no era posible acceder al listado de CAs reconocidas por el dispositivo móvil desde el interfaz gráfico de usuario, y mucho menos disponer de opciones en el interfaz de usuario que permitieran modificar y gestionar la lista de CAs raíz.
1184. Sin embargo, desde la versión 4.0 de Android ha sido posible acceder al repositorio de certificados de confianza, tanto de usuario como de sistema, desde la sección "Almacenamiento de Credenciales", disponible a través del menú "Ajustes [Personal] - Seguridad [Almacenamiento de Credenciales]", y en concreto, mediante la opción "Certificados de confianza".

⁸⁵ Es necesario analizar en detalle los componentes hardware del modelo de dispositivo móvil concreto para conocer las capacidades de almacenamiento seguro de credenciales disponibles a través del Keystore y del Keymaster.

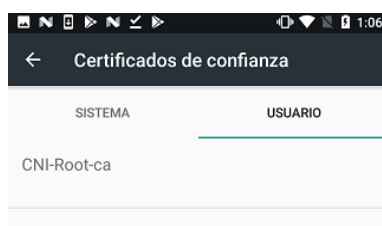
1185. Este repositorio de certificados de confianza también se conoce como el "Trust Store" [Ref.- 189] y es un elemento clave para la seguridad de Android.
1186. Esta sección dispone de dos pestañas, la primera de ellas asociada al "Sistema", es decir, a la lista de autoridades certificadoras (CAs) raíz reconocidas por defecto en esa versión concreta de Android, y la segunda asociada al "Usuario" (vacía inicialmente hasta que el usuario añada manualmente algún certificado de CA).
1187. Desde la pestaña "Sistema" es posible visualizar la lista completa de CAs de confianza existente por defecto, pudiendo ver todos los detalles de sus certificados digitales al seleccionar cualquiera de las CAs de la lista, así como desactivar de forma independiente cada uno de los certificados raíz de cada CA contenidos en este repositorio:



1188. Las CAs pueden ser deshabilitadas empleando el botón "Inhabilitar" situado en la parte inferior de la pantalla que muestra los detalles del certificado:



1189. En cualquier momento una CA deshabilitada previamente puede volver a ser habilitada, de nuevo empleando el botón "Habilitar" situado en la parte inferior de la pantalla que muestra los detalles del certificado.
1190. La pestaña "Usuario" no contiene ningún certificado digital hasta que no se importe manualmente un certificado de CA. Una vez se importa un certificado de CA (ver apartado "11.3. Gestión de certificados digitales"), los detalles del mismo están disponibles a través de esta pantalla. La pestaña "Usuario" sólo mostrará los certificados digitales de CAs añadidas por el usuario, pero no los certificados finales u hoja (ni de servidor, ni de cliente, personales o de usuario final):



1191. Mediante el SDK (Software Development Kit) de Android es posible inspeccionar y extraer manualmente desde un ordenador, a través del puerto USB, el conjunto de certificados raíz asociado a las autoridades certificadoras (CAs) existentes por defecto en el dispositivo móvil, accediendo a través de ADB (*Android Debug Bridge*).
1192. El directorio `/system/etc/security/cacerts` contiene un fichero por cada uno de los certificados digitales asociados a las CAs reconocidas por defecto por Android, en lugar de estar todos ellos contenidos en un único fichero contenedor "cacerts.bks" (BKS, BouncyCastle Key Store [Ref.- 27]), dentro del mismo directorio, tal como ocurría en versiones de Android previas a la versión 4.x:

```
shell@mobile:/system/etc/security $ ls -l
drwxr-xr-x root      root          1970-07-29 02:16 cacerts
```

```

-rw-r--r-- root      root      1931 2014-11-07 01:30 mac_permissions.xml
-rw-r--r-- root      root      1433 2014-07-29 02:33 otacerts.zip
shell@mobile:/system/etc/security $ cd cacerts
shell@mobile:/system/etc/security/cacerts $ ls -l
...
-rw-r--r-- root      root      5940 2014-11-07 01:30 fbl26c6d.0
-rw-r--r-- root      root      4692 2014-11-07 01:30 fde84897.0
-rw-r--r-- root      root      5136 2014-11-07 01:30 ff783690.0
...

```

1193. El nombre de cada fichero corresponde al *hash* del certificado digital que contiene [Ref.- 43], y mediante herramientas estándar, como por ejemplo "openssl", es posible añadir nuevos certificados (en dispositivos móviles Android *rooteados*).

1194. Para poder inspeccionar el contenido de cada uno de estos ficheros "*.0" es simplemente necesario abrirlos con un editor de texto, ya que contienen el certificado digital en formato PEM (base64), junto a la representación del mismo en ASCII para poder ver sus detalles:

```

shell@mobile:/system/etc/security/cacerts $ cat ff783690.0
-----BEGIN CERTIFICATE-----
MIIEEdCCAlygAwIBAgIQRL4MilAAJLQR0zYq/mUK/TANBgkqhkiG9w0BAQUFADCB
...
-----END CERTIFICATE-----
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      44:be:0c:8b:50:00:24:b4:11:d3:36:2a:fe:65:0a:fd
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=US, ST=UT, L=Salt Lake City, O=The USERTRUST Nerdware
    Validity
      Not Before: Jul  9 18:10:42 1999 GMT
      Not After : Jul  9 18:19:22 2019 GMT
...

```

1195. Por tanto, este directorio contiene la lista de todos los certificados digitales raíz con las CAs disponibles por defecto en Android.

1196. Desde el punto de vista de seguridad, se recomienda evaluar la necesidad de confiar en todas y cada una de las CAs disponibles por defecto en Android, gestionadas por empresas privadas de referencia en la industria, gobiernos, empresas privadas menos conocidas e institutos de investigación, y deshabilitar todas aquellas que no se consideran de confianza [Ref.- 192].

1197. Por ejemplo, en un estudio realizado sobre éstas, 15 CAs raíz todavía hacían uso de un certificado digital basado en una clave RSA de 1.024 bits, longitud considerada insuficiente actualmente (desde finales del año 2013⁸⁶), recomendándose al menos el uso de claves de 2.048 bits. Por otro lado, 6 CAs raíz todavía hacían uso de un certificado digital basado en una firma o *hash* obtenida mediante MD5 (considerado insuficiente desde hace años), mientras que 115 CAs raíz todavía hacían uso de firmas obtenidas mediante SHA-1 (considerado insuficiente más recientemente, recomendándose la utilización de SHA-256).

⁸⁶ <http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf>

1198. Adicionalmente, como se ha descrito previamente, desde Android 4.0 se extiende la posibilidad de añadir nuevos certificados al repositorio de certificados de confianza ("Trust Store") del sistema (certificados existentes por defecto) [Ref.- 188].
1199. En versiones anteriores de Android, los certificados de usuario se almacenaban bajo `"/data/misc/keychain"` (sólo accesible por "root" o "system").
1200. En Android 7.x, dado que el sistema operativo ofrece soporte multi-usuario, los certificados digitales añadidos por un usuario solo son accesibles para él. Por ello, tanto la pestaña correspondiente a los certificados de CA "Usuario" como el menú "Credenciales de usuario" solo mostrarán los que dicho usuario ha añadido.
1201. A nivel de sistema de ficheros, los certificados de CA importados por el usuario se almacenan bajo `"/data/misc/user/<id_de_usuario>"` en un subdirectorio "cacerts-added", y el subdirectorio "cacerts-removed" contiene los certificados digitales de sistema deshabilitados por el usuario (gestionados desde la pestaña "Sistema"), y complementan al directorio que almacena los certificados del sistema descrito previamente, `"/system/etc/security/cacerts"`.

```
root@mobile:/data/misc/user/0/cacerts-added # ls -l
-rw-r--r-- system system      883 2017-06-19 22:07 3d45599a.0
-rw-r--r-- system system      712 2017-06-20 14:01 7689a575.0
```

1202. Debido a que la partición de sistema (`"/system"`) donde se almacena el repositorio de certificados de confianza del sistema es de sólo lectura, cuando se deshabilita un certificado digital éste no puede ser eliminado físicamente del sistema de ficheros, sino que en su lugar se incluye en el directorio "cacerts-removed" bajo `"/data/misc/user/<id_de_usuario>"`:

```
root@mobile:/data/misc/user/0/cacerts-removed # ls -l
ls -l
-rw-r--r-- system system      1023 2017-06-22 11:35 84cba82f.0
-rw-r--r-- system system       979 2017-06-22 11:35 c3a6a9ad.0
```

1203. Por otro lado, desde Android 4.1 se extienden las capacidades de gestión de la confianza depositadas en el repositorio de certificados de confianza del sistema mediante listas negras (*blacklisting*) [Ref.- 190], que afectan tanto a apps como al navegador web y componentes WebView.

1204. Se dispone de dos listas negras de sistema para añadir las claves públicas de CAs comprometidas, y los números de serie de certificados digitales de entidades finales (End Entity, EE) comprometidos. Ambas listas son almacenadas tanto en la base de datos de la app "Ajustes", como en ficheros bajo `"/data/misc/keychain"` (sólo accesible por "root" o "system"):

```
/data/misc/keychain/pubkey_blacklist.txt - Contiene claves públicas o
certificados hoja (que no son de CA) revocados o comprometidos.

/data/misc/keychain/serial_blacklist.txt - Contiene los hashes o los
```

números de certificados de CAs comprometidas.

NOTA: En el dispositivo empleado para la elaboración de la presente guía estas listas estaban vacías y , por tanto, los ficheros no existían.

1205. Las implicaciones del uso de certificados digitales en funciones de seguridad críticas requieren que el propietario del dispositivo móvil sea muy cuidadoso a la hora de añadir nuevos certificados digitales al dispositivo móvil, y especialmente, al almacén de certificados raíz (o autoridades certificadoras reconocidas).

11.3. GESTIÓN DE CERTIFICADOS DIGITALES

1206. Android permite añadir al dispositivo móvil (importar) nuevos certificados digitales, tanto correspondientes a una autoridad certificadora (CA), raíz o intermedia, como certificados finales u hoja, entre los que se incluyen los certificados cliente o personales y los de servidor, complementando así los existentes por defecto.

1207. Las capacidades de gestión de certificados digitales de Android incluyen tanto la adición como la supresión de certificados digitales existentes por defecto (que también pueden ser deshabilitados temporalmente) o previamente importados.

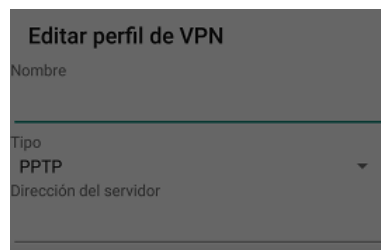
11.3.1. GESTIÓN DE CERTIFICADOS DIGITALES DE AUTORIDADES CERTIFICADORAS

1208. Los certificados digitales raíz asociados a una nueva CA pueden ser obtenidos (e importados) desde el almacenamiento del dispositivo, ya sea interno o externo, a través de la opción "Instalar desde almacenamiento" disponible desde el menú "Ajustes [Personal] - Seguridad [Almacenamiento de Credenciales]"⁸⁷.

1209. Para poder añadir un nuevo certificado de CA al repositorio de credenciales, es indispensable que el acceso al dispositivo esté protegido por un método de acceso. Si el dispositivo no tiene ninguno, al seleccionar el nombre del fichero que contiene el certificado, el sistema informará al usuario de que debe configurar un método de acceso, y llevará al menú de configuración de selección del código de acceso mediante el botón "Aceptar". Si dispone de él, se solicitará su introducción antes de proceder a la instalación del nuevo certificado.

1210. En Android 7.x, si se elimina el código de acceso del dispositivo móvil, se impedirá poder hacer uso del almacenamiento de credenciales, por ejemplo, al añadir un perfil VPN, de modo que no se podrá configurar dicho perfil en base a certificados que hayan sido añadidos previamente por el usuario mientras el código de acceso estaba configurado:

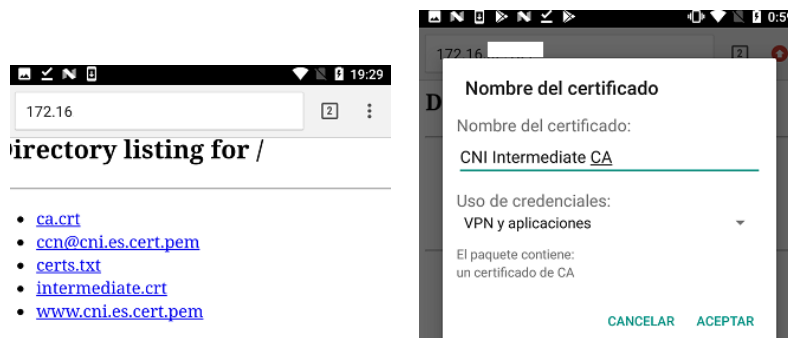
⁸⁷ En versiones previas de Android (2.x) el fichero del certificado digital debía almacenarse directamente en el directorio raíz del almacenamiento interno del dispositivo móvil (o tarjeta SD), no en un subdirectorio.

**Atención**

Debes establecer una contraseña o un PIN para la pantalla de bloqueo para usar el almacenamiento de credenciales.

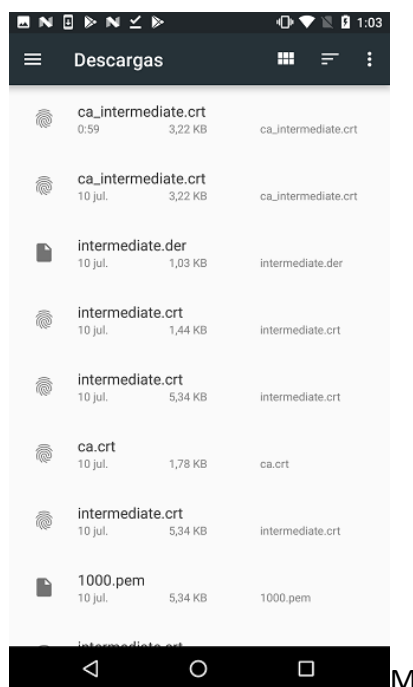
CANCELAR ACEPTAR

1211. Para instalar el certificado basta con transferir el fichero del certificado digital al dispositivo móvil, bien a través de un servidor web (preferiblemente propio, frente a servidores públicos de terceros [Ref.- 45]), correo electrónico, mediante una tarjeta de almacenamiento externa, dispositivo de almacenamiento USB OTG, o el almacenamiento interno del propio dispositivo móvil.
1212. Si el certificado digital es obtenido a través de un servidor web mediante el navegador web existente por defecto (Chrome), Android presentará al usuario directamente la ventana para importar el certificado digital en el dispositivo móvil⁸⁸:



1213. Adicionalmente, es posible descargar el certificado desde el navegador web, y almacenarlo en la carpeta "Descargas" (o "Download", dentro del almacenamiento interno). Los formatos reconocidos como certificado digital válido se detallan a continuación:

⁸⁸ Esta opción, disponible en versiones antiguas de Android, no estaba disponible en algunas versiones de Chrome en Android 6.x.



1214. Android 7.x admite certificados digitales X.509 codificados en formato DER (binario) siempre que tengan la extensión ".cer" o ".crt" (no siendo ".der" una extensión válida en Android) y también si están contenidos en un archivo de almacén de claves PKCS#12 con extensión ".p12" o ".pfx" [Ref.- 185]. Los archivos correspondientes a certificados admitidos por Android se representan con el icono de una huella dactilar a la izquierda.

1215. Al seleccionar el fichero correspondiente al certificado digital se solicitará al usuario un nombre para su identificación, y el propósito del mismo: "VPN y aplicaciones" o "Wi-Fi", tal como se mostró en las pantallas anteriores. Es importante reseñar que, si se va a utilizar un certificado para ambos usos (Wi-Fi, y VPN y aplicaciones), el certificado deberá importarse dos veces, una para cada escenario de uso.

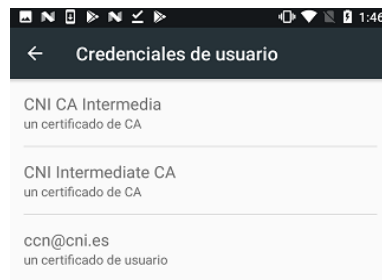
Nota: Los certificados raíz importados mediante estos métodos (desde Android 4.0) no sólo afectan a las conexiones Wi-Fi y a las redes VPN que hacen uso de certificados digitales, sino también a otro tipo de conexiones, como por ejemplo conexiones web autenticadas y cifradas mediante HTTPS, establecidas desde el navegador web, conexiones del cliente de correo electrónico, o cualquier otra app cliente, de ahí que la categoría para las redes VPN se denomine "VPN y aplicaciones" [Ref.- 189].

1216. El proceso de instalación del nuevo certificado digital no muestra los detalles del mismo, por lo que es muy importante ser especialmente cuidadoso a la hora de importar nuevos certificados, especialmente los descargados directamente desde la web.

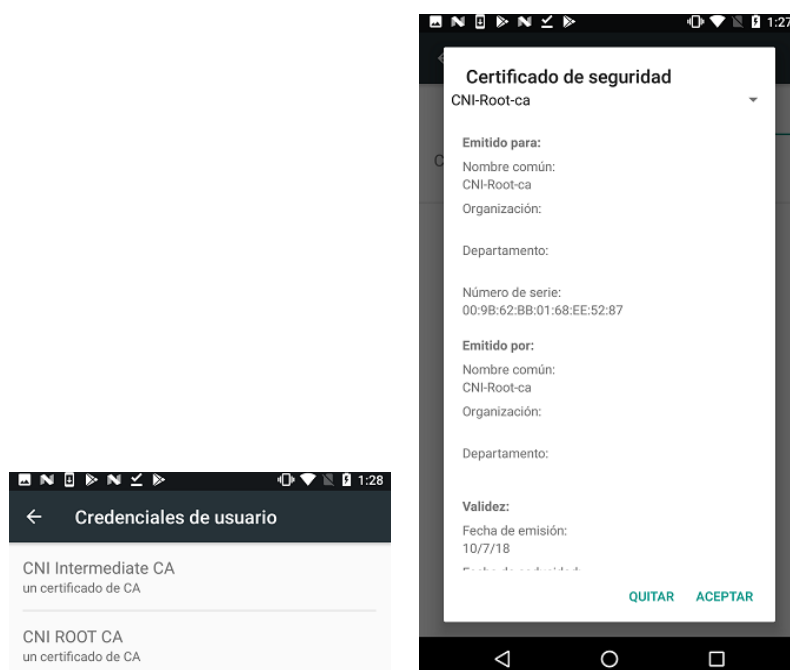
1217. Si se instala un certificado digital correspondiente a una CA intermedia, el certificado de la CA raíz no será añadido al almacén de credenciales.

1218. Android 7.x ofrece como novedad la opción "Credenciales de usuario" en el menú "Ajustes - [Personal] Seguridad [Almacenamiento de credenciales]", que permite al usuario obtener la lista de los certificados importados por él

correspondientes a CAs o a certificados de usuario, cliente o personales, en ambos casos, para uso de "VPN y aplicaciones".

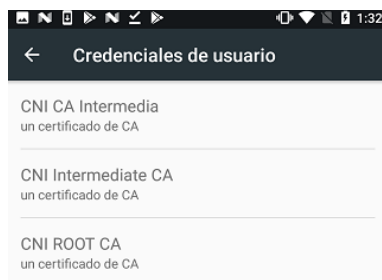


1219. El proceso para añadir certificados digitales raíz específicos para su utilización en conexiones Wi-Fi también se puede realizar a través de la opción "Ajustes - Wi-Fi - [...] Ajustes avanzados - Instalar certificados".
1220. Sin embargo, en Android 7.x no es posible visualizar los certificados importados para "Wi-Fi", ni en la sección "Credenciales de usuario" ni en la sección "Certificados de confianza - Usuario". Esto provoca que no sea posible gestionarlos individualmente, por lo que no se pueden eliminar a menos que se resetee el almacén de credenciales mediante la opción "Eliminar certificados".
1221. En el menú "[Almacenamiento de credenciales] Certificados de confianza - Usuario" se presentan los certificados correspondientes a CAs que se hayan añadido al sistema, ya sean CAs intermedias o CAs raíz, para "VPN y aplicaciones". Sin embargo, este menú no incluye los certificados cliente o personales importados por el usuario.
1222. El nombre que se visualiza en la pestaña "Usuario" del menú "Certificados de Confianza" para una CA raíz será el indique el campo "Common Name" (CN) definido en el propio certificado, y no el nombre introducido por el usuario durante su importación. Mediante la selección del certificado digital es posible ver todos sus detalles:



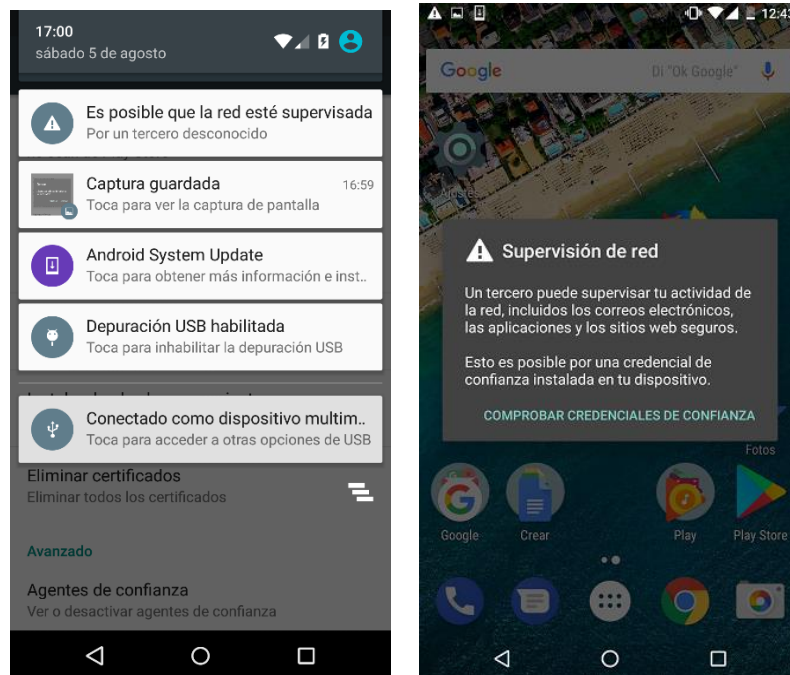
Nota: Android permite importar un mismo certificado digital raíz en múltiples ocasiones, pero en el listado de certificados de confianza, pestaña "Usuario", sólo aparecerá una vez dicho certificado digital, identificado por su "Common Name" (CN) y, por tanto, independientemente del nombre proporcionado al importarlo cada vez (que puede ser igual o diferente), a diferencia de en el menú "Credenciales de usuario".

1223. El nombre que el usuario defina para el certificado durante su importación se presenta en el menú "Credenciales de usuario" y en el menú de configuración de Wi-Fi y de VPN. Si se realiza la importación de un mismo certificado más de una vez empleando distintos nombres, en el menú "Credenciales de usuario" aparecerán tantas entradas como nombres diferentes se hayan empleado para la importación del certificado, independientemente del correspondiente al CN del mismo:



1224. En el menú de configuración de redes Wi-Fi empresariales se mostrarán tantas entradas para el certificado como veces se haya importado el mismo. Cada una de las instancias del certificado digital aparecerá con el nombre proporcionado al importarlo, y en el caso de duplicar dicho nombre durante el proceso de importación, se presenta únicamente una entrada.
1225. Para eliminar un certificado digital correspondiente a una CA de manera selectiva, se puede acceder al menú "[Almacenamiento de credenciales] Certificados de confianza - Usuario", ver los detalles del mismo, pudiendo hacer uso del botón "Quitar" disponible en la parte inferior.
1226. Si se selecciona el nombre de una CA a través del menú "[Almacenamiento de credenciales] - Credenciales de usuario" y se pulsa el botón "Quitar", el certificado desaparecerá de dicho menú, pero el certificado de CA seguirá presente en la sección de "[Almacenamiento de credenciales] Certificados de confianza - Usuario".
1227. El certificado digital es importado en la pestaña "Usuario", aunque se trate de un certificado digital correspondiente a una CA (el término puede resultar confuso, ya que no hace referencia a certificados digitales de usuario o personales), ya que ha sido añadido a posteriori por el usuario y no forma parte de la configuración por defecto del sistema (pestaña "Sistema").
1228. En Android 4.x, tras llevar a cabo la instalación de un certificado digital asociado a una CA, la sección "Seguridad" del menú "Ajustes" muestra un triángulo de aviso. Asimismo, se muestra al usuario de manera permanente una notificación a través de la barra superior de menú. Al seleccionar el triángulo se informa al usuario de que se ha instalado una nueva CA, lo que puede permitir a un tercero supervisar las actividades del usuario mediante la interceptación de su tráfico de red cifrado.

1229. En Android 5.x y Android 6.x, desapareció el aviso de triángulo de la sección "Seguridad" del menú "Ajustes", y se genera el aviso con un icono de triángulo sólo como parte de una notificación que no se puede descartar. Al seleccionar la misma, se indicará que la red puede estar supervisada:



1230. Esta notificación no se mostraba si el certificado de la CA sólo se había importado como parte de un certificado cliente (consultar apartado "11.3.3. Gestión de certificados digitales cliente").
1231. En Android 7.x este aviso no se muestra, potencialmente debido a que los certificados de CA instalados por el usuario, y los instalados a través de APIs de administración de dispositivos móviles, ya no se consideran de confianza por defecto para las apps que implementan la API de nivel 24 en adelante, por lo que la negociación HTTPS fallará si se recibe un certificado de este tipo.

11.3.2. GESTIÓN DE CERTIFICADOS DIGITALES SERVIDOR

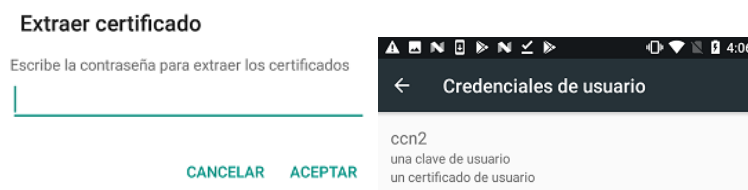
1232. Los certificados digitales de servidor se importan de forma similar a los certificados correspondientes a una autoridad certificadora, tal como se detalla en el apartado anterior.
1233. Aunque hasta Android 6.x no existía la posibilidad de visualizar este tipo de certificados, en Android 7.x se muestran en el menú "Ajustes - [Personal] Seguridad - [Almacenamiento de credenciales] - Credenciales de usuario".
1234. Desde dicho menú, seleccionando el certificado correspondiente seguido de la opción "Quitar" es posible eliminar certificados de servidor importados previamente por el usuario.

11.3.3. GESTIÓN DE CERTIFICADOS DIGITALES CLIENTE

1235. Los certificados digitales cliente (personales o de usuario⁸⁹) asociados, por ejemplo, a redes VPN, servidores web o redes Wi-Fi, pueden ser importados desde el almacenamiento (interno o externo) del dispositivo, a través de la opción "Instalar desde almacenamiento" disponible desde el menú "Ajustes - [Personal] Seguridad - [Almacenamiento de Credenciales] Credenciales de usuario".
1236. Además, para el caso de las redes Wi-Fi empresariales, es posible añadir el certificado cliente a través del menú de configuración de la red Wi-Fi ("Ajustes - [Conexiones inalámbricas y redes] Wi-Fi - <...> Ajustes avanzados - Instalar certificados"):



1237. Al igual que en el caso de los certificados digitales raíz (ver apartado "11.3.1. Gestión de certificados digitales de Autoridades Certificadoras"), Android solicitará al usuario establecer una contraseña o código de acceso en el dispositivo móvil para poder hacer uso del repositorio de credenciales, si aún no se ha establecido una.
1238. Para archivos que actúan como almacén de claves, Android soporta certificados X.509 codificados como PKCS#12. Estos ficheros deberán tener extensión .p12 o .pfx [Ref.- 185]. Al seleccionar el fichero correspondiente al certificado digital cliente se solicitará al usuario la contraseña que fue empleada para proteger el fichero PKCS#12 que contiene el certificado digital y la clave privada, y poder así extraer los certificados del fichero:



1239. Si el fichero del certificado corresponde a un archivo de certificado estándar (que, por tanto, no incluye clave privada), el texto dentro del campo "El paquete contiene" sólo incluirá "un certificado de usuario":

⁸⁹ En este apartado se hace uso del término certificado digital cliente (o personal) para referirse a los certificados empleados para la autenticación del usuario, en lugar de emplearse el término certificado digital de usuario, con el objetivo de evitar confusiones con la pestaña "Usuario" de Android, que muestra tanto certificados de cliente como de CAs (tal y como se ha descrito en apartados previos).

Nombre del certificado

Nombre del certificado:

Uso de credenciales:
VPN y aplicaciones

El paquete contiene:
un certificado de usuario

CANCELAR ACEPTAR

1240. Si el fichero correspondiente al certificado fuese un almacén de claves y contuviese además la clave privada del cliente, Android mostrará el texto "una clave de usuario" y "un certificado de usuario" dentro del campo "El paquete contiene", como se ilustró en la imagen correspondiente a la importación de un fichero ".pfx" anteriormente.
1241. Si el fichero correspondiente al certificado contuviese además la clave pública de la CA que firmó dicho certificado, Android mostrará el texto "*un certificado de CA*" y "*un certificado de usuario*" dentro del campo "El paquete contiene". Este escenario resulta útil para evitar tener que añadir primero el certificado de la CA y luego el del cliente. Los ficheros que contienen un certificado de usuario deben estar en formato PKCS#12.
1242. Aunque hasta Android 6.x inclusive no existía una opción que permitiese consultar los certificados cliente instalados en el dispositivo móvil, no siendo posible eliminarlos de forma selectiva, sino todos en bloque, el menú "Credenciales de usuario" de la sección "[Almacenamiento de credenciales]" de Android 7.x sí permite al usuario obtener la lista de los certificados cliente importados por él para uso de "VPN y aplicaciones", no siendo posible ver los certificados cliente añadidos para Wi-Fi desde dicho menú.
1243. Tras extraer el certificado digital cliente (que contiene la clave pública), y la clave privada asociada (y opcionalmente, el certificado de la CA que lo ha emitido), se solicitará al usuario un nombre para su identificación y el propósito del mismo: "VPN y aplicaciones" o "Wi-Fi". Aunque en Android 6.x el nombre asignado al certificado por defecto correspondía al denominado "*friendlyName*", en Android 7.x se vuelve al modelo empleado en versiones anteriores de Android (5.x y 4.x) basado en el identificador local de la clave, "*localKeyID*" como nombre mostrado por defecto para el certificado:

Nombre del certificado

Nombre del certificado:

2867a07eb414dcc530845d5ea75762f4c

Uso de credenciales:
VPN y aplicaciones

El paquete contiene:
una clave de usuario
un certificado de usuario

CANCELAR ACEPTAR

Credenciales de usuario

ecf71292867a07eb414dcc530845d5ea75762f4c
una clave de usuario
un certificado de usuario

Detalles de credenciales

ecf71292867a07eb414dcc530845d5ea75762f4c
una clave de usuario
un certificado de usuario

QUITAR LISTO

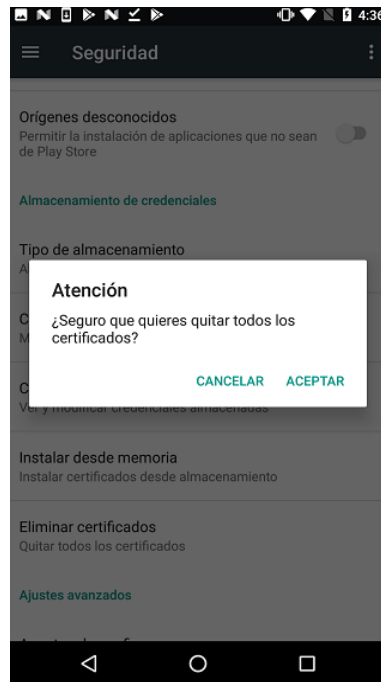

```
$ openssl pkcs12 -info -in ccn2@cni.es.pfx
Enter Import Password:
MAC:sha1 Iteration 2048
PKCS7 Encrypted data: pbeWithSHA1And40BitRC2-CBC, Iteration 2048
Certificate bag
Bag Attributes
    localKeyID: EC F7 12 92 86 7A 07 EB 41 4D CC 53 08 45 D5 EA 75 76 2F 4C
subject=/C=ES/ST=Madrid/O=CNI/CN=CNI/emailAddress=info@cni.es
issuer=/CN=CNI-Root-ca
...
```

1244. Se recomienda proporcionar un nombre de certificado que permita la identificación del mismo de forma sencilla desde las aplicaciones cliente que hacen uso de éste. Este nombre será mostrado posteriormente al seleccionar el certificado digital cliente, por ejemplo, en el proceso de configuración de una red Wi-Fi empresarial o de una red VPN.

Nota: Android permite importar un mismo certificado digital cliente en múltiples ocasiones, apareciendo múltiples veces en el listado de certificados digitales cliente (o de usuario) disponible a la hora de hacer uso del mismo, por ejemplo, al configurar una red Wi-Fi empresarial. Cada una de las instancias del certificado digital aparecerá con el nombre proporcionado al importarlo, y en el caso de duplicar dicho nombre durante el proceso de importación, se presenta únicamente una entrada.

1245. Si se incluye un certificado de una CA junto con el certificado digital cliente, éste se instalará de forma simultánea. El certificado digital correspondiente a la CA emisora del certificado digital cliente, si está incluido en el fichero PKCS#12, es también importado en la pestaña "Usuario"⁹⁰, al tratarse de un certificado digital correspondiente a una CA, y que ha sido añadido a posteriori por el usuario (y que por tanto no forma parte de la configuración por defecto del sistema, pestaña "Sistema").
1246. Para eliminar un certificado de forma individual se puede seleccionar la opción del menú "Ajustes - [Personal] Seguridad - [Almacenamiento de credenciales] Credenciales de usuario - <Marcar el certificado> - Quitar".
1247. Para eliminar todos los certificados existentes en el repositorio de credenciales que hayan sido añadidos por el usuario, lo que equivale a resetear el repositorio de credenciales para volver a hacer uso del mismo desde la versión de fábrica (formado por las CAs reconocidas por Android), se dispone de la opción "Ajustes - [Personal] Seguridad - [Almacenamiento de credenciales] Eliminar certificados":

⁹⁰ A diferencia de como ocurría en versiones previas de Android (2.x), el certificado digital importado (ya sea de una CA o personal) no es eliminado de la tarjeta de almacenamiento externa una vez se ha completado la operación de importación.



1248. Los certificados digitales cliente que se importen con su clave privada podrán ser utilizados en aquellas apps que requieran autenticación mediante este método, como por ejemplo, al acceder con un navegador web a un servicio web que requiere disponer de un certificado de usuario.

11.4. CERTIFICATE PINNING Y CERTIFICATE BLACKLISTING

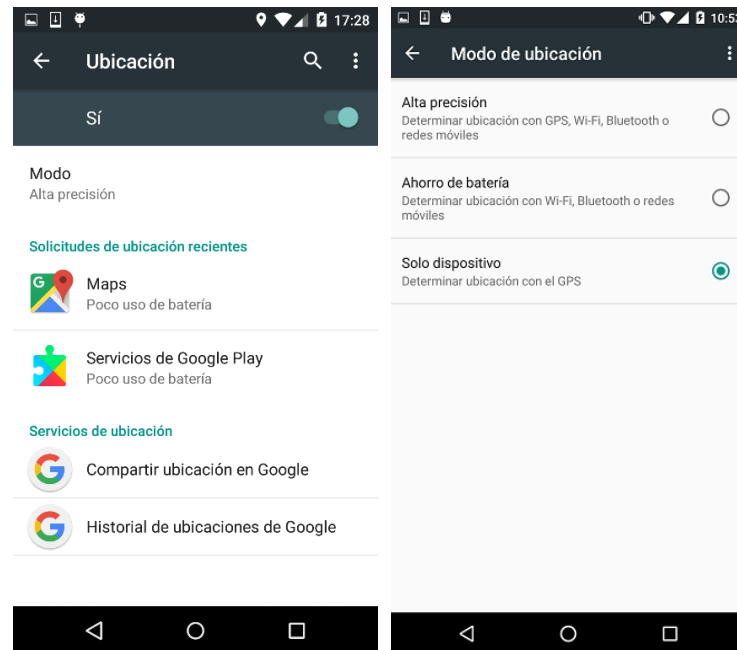
1249. Android, desde la versión 4.2 (API 17), dispone de capacidades adicionales para el uso de mecanismos de validación y protección de las comunicaciones autenticadas y cifradas mediante HTTPS (TLS), basados en *certificate pinning* [Ref.- 79].
1250. El mecanismo de *certificate pinning* (en adelante, CP) consiste en una verificación del certificado del servidor en el lado cliente. Para que la verificación sea posible, el certificado del servidor o su huella digital deben ser previamente conocidos por la app (o el software cliente), es decir, debe disponer del pin. De este modo, cuando se establezca la conexión con el servidor, la app podrá comparar la huella (o pin conocido) con el certificado recibido desde el servidor remoto. Si son idénticos, la conexión se considerará válida y la transferencia de datos se iniciará. Si no lo son, la app deberá interrumpir la negociación TLS y rechazar la conexión.
1251. En versiones anteriores de Android, para implementar CP en una app existían principalmente dos mecanismos:
- OkHttp y CertificatePinner (entre otros): mediante librerías específicas que requieren que la app disponga de los certificados (o sus huellas digitales) como parte de su código, o de sus recursos o ficheros asociados, y es posible añadir múltiples huellas para dominios y/o certificados diferentes para flexibilizar la funcionalidad de la app.
 - TrustManager: mediante la implementación personalizada de la clase estándar TrustManager de Java, que también requiere disponer de los

certificados (o sus huellas digitales) como parte del código o recursos de la app.

1252. Con la nueva característica conocida como "*Network security configuration (NSC)*" introducida por Android 7.x [Ref.- 747] es posible declarar los métodos de comunicación que utilizará la app, incluido CP, utilizando ficheros de configuración XML (en los que se definen las huellas digitales de los certificados o pines) y utilizando el atributo *networkSecurityConfig* en la etiqueta "<application>", lo que simplifica mucho la implementación de CP para los desarrolladores de las apps.
1253. El mecanismo de *blacklisting* (lista negra) se emplea para proteger al sistema tanto de CAs que puedan haber sido comprometidas, como de certificados digitales o claves públicas que pueden haberse revocado o comprometido.
1254. En el pasado, ya ha habido casos de gran impacto que revelan que, aunque comprometer una CA no es habitual, demuestran que es posible y un escenario de ataque a tener en cuenta [Ref.- 267].
1255. Para mitigar el riesgo, Android permite añadir a una lista negra tanto certificados como claves públicas, incluso asociados a una CA. Desde Android 4.2 esta lista, que se compone de dos ficheros, puede ser actualizada de forma remota para añadir nuevas entradas (ver apartado "11.2. Certificados digitales de sistema y de usuario").

12. LOCALIZACIÓN (O UBICACIÓN) GEOGRÁFICA

1256. Las capacidades de localización (o ubicación) del dispositivo móvil a través del módulo de GPS, o de las redes de datos (telefonía móvil y Wi-Fi), pueden ser activadas y desactivadas por el usuario en función de su utilización.
1257. El dispositivo o módulo GPS en Android está activo únicamente cuando se está haciendo uso de una app con capacidades de localización, es decir, que hace uso del GPS, y en base a la configuración específica (modos), de los servicios de localización de Android.
1258. En ese momento, y si el servicio de ubicación está activo, se mostrará un icono en la parte derecha de la barra de estado similar a una gota de agua invertida.
1259. Si ninguna app hace uso del servicio de localización en ese instante, aunque la ubicación esté activa no se mostrará ningún indicador en la barra de estado. Ello dificulta que el usuario sea consciente de esa circunstancia.
1260. Las capacidades de localización del dispositivo móvil se habilitan a través del interruptor "[SI | NO]" desde el menú "Ajustes [Personal] - Ubicación".



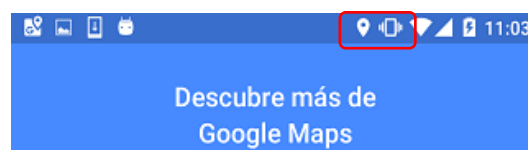
1261. Desde Android 4.4 (KitKat) se proporciona un mayor control sobre los diferentes modos de localización (o ubicación) disponibles [Ref.- 149]: alta precisión, ahorro de batería o solo en dispositivo.

1262. Las opciones son:

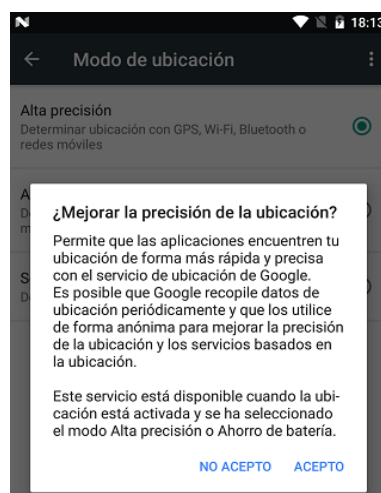
- "Alta precisión": hace uso del módulo GPS, de las conexiones Bluetooth (las cuales no se empleaban para el servicio de ubicación en Android 5.x y anteriores), de las redes Wi-Fi y de telefonía móvil, y de otros sensores, para obtener la ubicación más rápida y precisa posible en el dispositivo móvil.
- "Ahorro de batería": emplea como fuentes para la obtención de la ubicación del dispositivo móvil las redes Wi-Fi, las conexiones Bluetooth y las redes de telefonía móvil, ya que llevan a cabo un menor consumo de energía en comparación al módulo GPS.
- "Solo en dispositivo": hace uso únicamente del módulo GPS del dispositivo móvil.

1263. La nueva funcionalidad y API empleada por las apps para acceder a los servicios de localización, especialmente al usar el modo "Ahorro de batería", reduce drásticamente el consumo de batería. Previamente, cada app que necesitaba hacer uso de los servicios de localización despertaba al módulo GPS hardware y obtenía su información de ubicación propia individualmente.

1264. Cuando una aplicación o servicio del dispositivo móvil interactúa con los servicios de ubicación, se mostrará en la barra de ajustes rápidos el icono asociado, que aparece a continuación junto al símbolo del modo vibración:



1265. Todos estos modos están disponibles en los servicios de ubicación de Google, que extienden las capacidades del módulo GPS del dispositivo móvil con otras tecnologías.
1266. La opción (o modo) habilitada por defecto dependerá de las opciones seleccionadas en la pantalla de "Ubicación y Google" durante el proceso inicial de instalación y configuración del dispositivo móvil (ver apartado "23. Apéndice A: Proceso de instalación y configuración inicial").
1267. La activación del modo avión de Android deshabilitará los servicios de ubicación, no permitiendo ninguna transmisión inalámbrica ni con los satélites GPS, ni con las redes Wi-Fi o de telefonía móvil, ni con dispositivos Bluetooth.
1268. Desde el punto de vista de seguridad, si se desea que no se genere ningún tráfico desde el dispositivo móvil hacia los servicios de Google, se recomienda seleccionar la opción o modo "Solo en dispositivo", pese a que tenga asociado un mayor consumo de batería y tarde más tiempo en localizar la ubicación del dispositivo inicialmente.
1269. En el caso de habilitar las capacidades de localización a través de redes y conexiones inalámbricas (Wi-Fi, Bluetooth y telefonía móvil), mediante la selección de los modos "Alta precisión" o "Ahorro de batería", Android solicitará confirmación por parte del usuario para recopilar los datos de las señales inalámbricas recibidas y la ubicación del terminal de forma anónima:



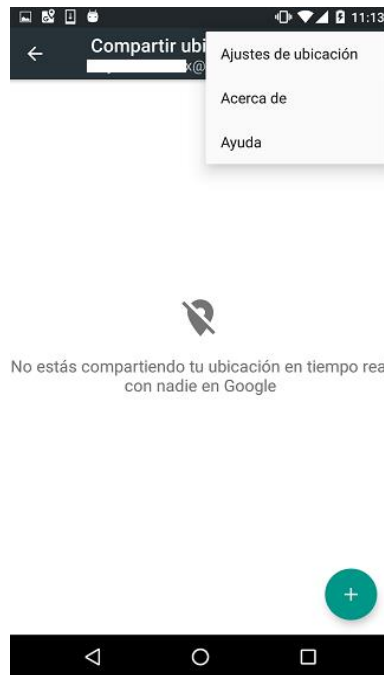
Nota: Existen estudios pasados que reflejan que la información de localización intercambiada entre el dispositivo móvil Android y Google no es completamente anónima, ya que, aparte de incluir detalles de las señales de las redes inalámbricas recibidas y la ubicación del terminal, se incluye un identificador único del dispositivo móvil [Ref.- 82], lo cual puede afectar a la privacidad del usuario.

Adicionalmente, la información de las redes Wi-Fi recopilada por Google a través de los coches del programa StreetView y de los dispositivos móviles Android estuvo disponible en el pasado para su consulta a través de la API de Google y de páginas web externas [Ref.- 83], siendo posible conocer la ubicación de una red Wi-Fi a nivel mundial en base a su dirección MAC o BSSID. El acceso a través de este servicio no está disponible actualmente de manera pública.

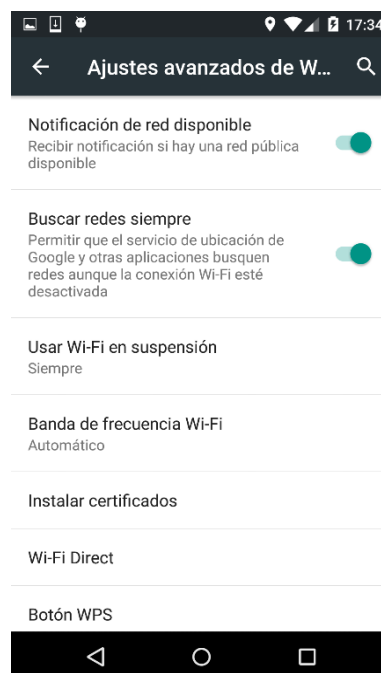
1270. Es posible consultar las apps que han solicitado información sobre la ubicación del dispositivo móvil recientemente a través del menú "Ajustes [Personal] - Ubicación", y de la sección "Solicitudes de Ubicación Recientes".



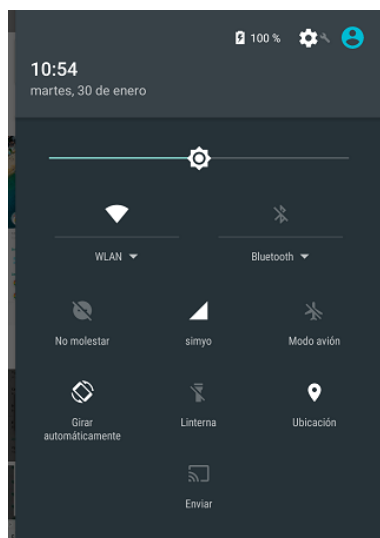
1271. Mediante la selección de una app de la lista se accede a la información de la app (información también disponible desde el menú "Ajustes [Dispositivo] - Aplicaciones - Todas"), incluyendo su uso de batería, permisos, uso de almacenamiento, etc.
1272. Adicionalmente, una vez se ha configurado una cuenta de usuario de Google en el dispositivo móvil, a través del menú "Ajustes [Personal] - Ubicación", y de la sección "Servicios de Ubicación", es posible seleccionar "Historial de ubicaciones de Google" y acceder a los ajustes de configuración para activar o desactivar los informes de ubicación y el historial de ubicaciones de la cuenta de usuario de Google (ver apartado "12.5. Historial de ubicaciones").
1273. La opción "Ajustes - [Personal] Ubicación - [Servicios de Ubicación] Compartir ubicación en Google" permite ver con quién se está compartiendo la ubicación en un momento concreto.
1274. La opción de "Compartir ubicación" se habilita en el menú lateral de la aplicación "Google Maps". Esta opción requiere disponer de una conexión de datos.
1275. Cuando se selecciona, aparecerá un menú que permite especificar el tiempo durante el cual se compartirá la ubicación, así como los contactos con los que se desea compartir (que deben poseer también una cuenta de Google). Es posible incluso crear un enlace que se puede enviar a través de aplicaciones de mensajería, Bluetooth, etc., de modo que los contactos elegidos o que reciban el enlace puedan realizar el seguimiento en tiempo real de la posición del dispositivo en un mapa. Por este motivo y debido a las implicaciones en la privacidad, una vez se comparte la ubicación con alguien, el usuario recibirá una notificación informándole de ello a modo de recordatorio. Para más detalles, consultar el apartado "12.2. Google Maps".



1276. Adicionalmente, si se hace uso de alguno de los modos que emplean las redes Wi-Fi para obtener información sobre la ubicación del dispositivo móvil, es posible configurar si se permite el uso de las capacidades Wi-Fi del dispositivo móvil para estos servicios y, en concreto, para la búsqueda e identificación de redes Wi-Fi, aunque la conexión o el interfaz Wi-Fi esté desactivado.
1277. Esta opción de configuración también está disponible en el proceso de instalación y configuración inicial del terminal, ver apartado "23. Apéndice A: Proceso de instalación y configuración inicial"), y se puede acceder a la misma posteriormente desde el menú "Ajustes - [Conexiones Inalámbricas y Redes] Wi-Fi - [...] - Ajustes avanzados - Buscar redes siempre":



1278. Por último, la activación de los servicios de ubicación en el dispositivo móvil también está disponible a través de los ajustes rápidos, accesibles desde la barra superior de estado:



1279. Para que el servicio "Encontrar mi dispositivo" (consultar apartado "8.1. Gestión remota de dispositivos Android por parte del usuario") esté operativo, es requisito indispensable que las capacidades de ubicación estén activas.
1280. Se recomienda valorar el uso de las capacidades de localización del dispositivo móvil, buscando el equilibrio entre privacidad y funcionalidad. En caso de querer hacer uso de sus capacidades y evitar otras implicaciones de privacidad, se recomienda activar únicamente la localización mediante GPS, no haciendo uso de las capacidades de localización mediante redes Wi-Fi o redes de telefonía móvil.
1281. En noviembre de 2017, la empresa Quartz publicó un informe en el que desvelaba que, desde el comienzo de dicho año, todos los dispositivos móviles Android recolectan información sobre la ubicación de las torres de telefonía móvil cercanas, y que envían esos datos a Google cuando se conectan a Internet, incluso aunque los servicios de ubicación hayan sido deshabilitados por el propietario del terminal. Google confirmó esta información alegando que los datos no se almacenaban, sino que se empleaban para mejorar el servicio de mensajes y notificaciones. En el momento de elaboración de la presente guía, no existe ninguna opción para deshabilitar este comportamiento en el dispositivo móvil, ni se incluye referencia alguna a esta práctica en la política de privacidad de Google [Ref.-252].

12.1. SERVICIOS DE UBICACIÓN DE GOOGLE

1282. Durante el proceso de configuración inicial del dispositivo móvil empleado para la elaboración de la presente guía, Android solicita al usuario si desea usar los servicios de ubicación de Google, incluso aunque se haya declinado la asociación del dispositivo a una cuenta de usuario de Google.

1283. Las opciones disponibles "Usar el servicio de ubicación de Google" y "Ayudar a mejorar los servicios de ubicación" están habilitadas por defecto, por lo que se recomienda desactivarlas hasta determinar el uso que se desea hacer de ellas (ver apartado "23. Apéndice A: Proceso de instalación y configuración inicial").
1284. Los servicios de ubicación o localización de Google (*Google Location Services*), permiten a las apps conocer la ubicación geográfica aproximada del dispositivo móvil, y por tanto del usuario, aun sin disponer de señal GPS, y hacer uso de esa información en los servicios de búsqueda (y otros servicios) de Google, o desde cualquier app con los permisos adecuados, teóricamente de una forma más rápida y precisa.
1285. La ubicación se obtiene a través de la información de localización disponible sobre torres (o redes) de telefonía móvil y redes Wi-Fi, y la señal recibida en cada momento por el dispositivo móvil para ambas tecnologías (telefonía móvil y Wi-Fi), en lugar de emplear información más precisa como la proporcionada por el dispositivo o módulo GPS.
1286. Para hacer uso de la geolocalización en base a redes Wi-Fi, el dispositivo escaneará los puntos de acceso Wi-Fi que estén en su rango de alcance (independientemente de su nivel de seguridad) y, en base a las respuestas recibidas, y tras enviar los detalles a los servicios de Google, la respuesta obtenida por parte de estos le permitirá interpolar su ubicación aproximada.
1287. El servicio de ubicación a través de Bluetooth se basa en el uso de las denominadas balizas Bluetooth LE y la tecnología conocida como *beacons*. Un *beacon* es un dispositivo inalámbrico que envía periódicamente un anuncio BLE (*Bluetooth Low Energy*) con su identificador, de forma que una app pueda extraer su ubicación a través de una base de datos con la posición de estas balizas. El dispositivo móvil recibe ese anuncio y la app (o el servicio de ubicación del sistema operativo) puede determinar así la ubicación aproximada.
1288. Las dos principales (aunque no las únicas) tecnologías de *beacons* que existen actualmente en el mercado son iBeacon (de Apple) [Ref.- 253] y EddyStone (de Google) [Ref.- 254].
1289. La utilización de los servicios de ubicación de Google puede conllevar por tanto el envío, supuestamente anónimo, a Google de los datos recopilados sobre las redes Wi-Fi, de telefonía móvil y de dispositivos Bluetooth basados en tecnologías basadas en *beacons* sobre la ubicación aproximada en la que se encuentra el dispositivo móvil, incluso aunque no se esté utilizando ninguna app.
1290. El servicio Google Nearby (ver apartado "15.9. Google Nearby") también realiza acciones de localización, utilizando escaneos Wi-Fi, Bluetooth y BLE, incluso aunque ambas interfaces hayan sido deshabilitados a propósito por el usuario.
1291. En resumen, estas peticiones de información hacia Google pueden desvelar información de las redes Wi-Fi, de telefonía móvil y de las balizas BLE existentes en el área de cobertura del dispositivo móvil, así como otros

detalles relevantes del terminal, potencialmente a cualquiera que esté capturando el tráfico de datos generado desde el dispositivo móvil. Las respuestas de los servicios de ubicación de Google, en formato binario, contienen información de la ubicación estimada del dispositivo móvil.

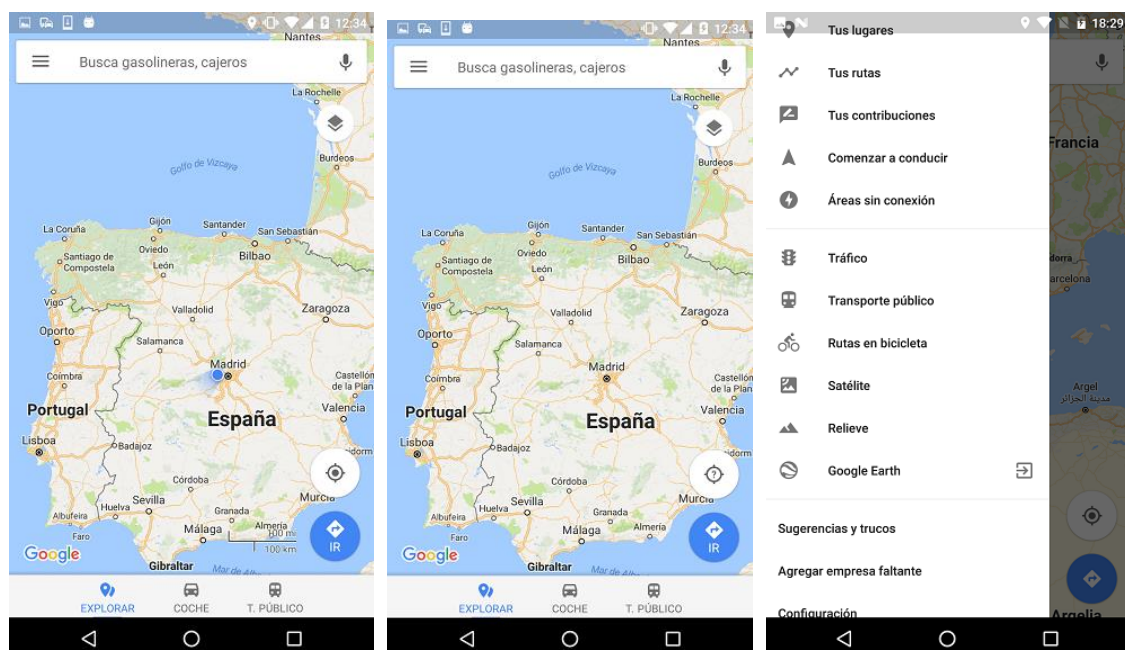
1292. Se recomienda limitar el uso de estos servicios, especialmente a través de redes de datos inseguras, por parte de cualquier app como por ejemplo "Google Maps" (ver siguiente apartado), y específicamente su utilización cuando se emplea algo más que el módulo GPS, ya que los servicios de ubicación de Google a través de torres de telefonía móvil y redes Wi-Fi desvelan detalles privados del dispositivo móvil y de su ubicación.

12.2. GOOGLE MAPS

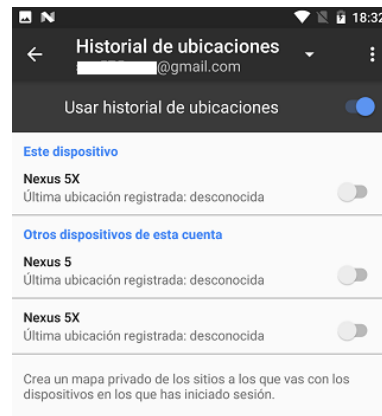
1293. Los servicios de ubicación de Google son utilizados, entre otros, tanto por el propio sistema operativo Android como por las apps disponibles por defecto o proporcionadas por terceros, como por ejemplo la app "Maps" o "Google Maps" (<https://maps.google.com>) que, en el caso del dispositivo móvil empleado para la elaboración de la presente guía, está disponible como app por defecto a través de la app "Maps" (disponible desde el *Launcher*).

1294. Tras aceptar los términos y condiciones del servicio (al ejecutar la app la primera vez), y en función de la configuración de los servicios de ubicación, la recopilación de información será activada y se permitirá su utilización por defecto por parte de las apps que hacen uso de los servicios de ubicación.

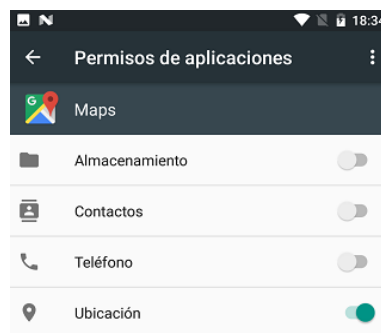
1295. Desde Maps es posible hacer uso de la información de localización, en concreto, empleando el botón circular con una cruceta, que centrará el mapa en la localización actual del usuario (identificada por un indicador azul). Si el indicador de posición se muestra como un punto gris, significa que Maps no consigue detectar la ubicación actual y está mostrando la última ubicación que determinó anteriormente para el dispositivo móvil.



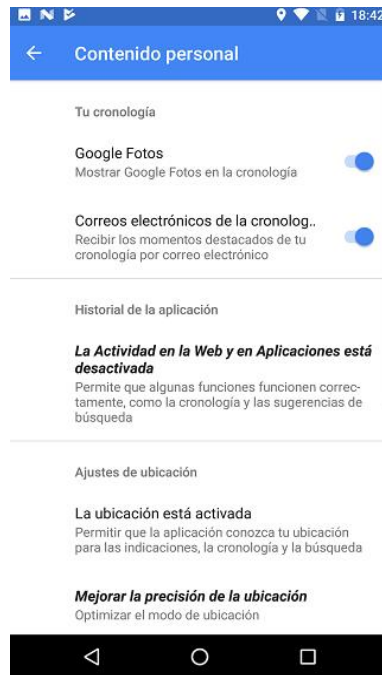
1296. La versión de la aplicación Maps se puede consultar dentro del menú de la app, "Ajustes - Información, condiciones y privacidad".
1297. Al hacer uso de apps que requieren conocer la ubicación del usuario, como Maps, y si han sido habilitadas, se emplearán las capacidades para localizar la ubicación del dispositivo móvil incluso cuando no se disponga de módulo GPS (o éste no esté activo o no haya cobertura de los satélites GPS), a través de A-GPS, que emplea la información de localización disponible a través de servicios de terceros (ver apartado "12.3. A-GPS"), y la funcionalidad extendida de los servicios de ubicación de Google descritos previamente, haciendo uso de la información de las redes Wi-Fi, las balizas BLE y las torres de telefonía móvil.
1298. Los servicios de ubicación de Google integrados en Google Maps, así como los datos e información obtenida por la app (mapas, información extra, etc.), en el caso de la versión 9.82.2 de la app Maps (empleada en la elaboración de la presente guía) emplean únicamente conexiones mediante HTTPS (cifradas) desde el dispositivo móvil hacia múltiples servidores de Google. Versiones pasadas de los servicios de localización de Google enviaban tráfico no cifrado mediante HTTP.
1299. Este cambio en el funcionamiento respecto al comportamiento de versiones previas de la app y versiones previas de Android evita que un potencial atacante acceda fácilmente al tráfico de red y pueda obtener detalles de la ubicación del usuario y de las redes inalámbricas (como por ejemplo el nombre de red, SSID, y su dirección MAC, BSSID) y de las torres de telefonía móvil (como por ejemplo el identificador de torre, y los códigos de área (LAC), operador de telefonía móvil (MNC) y país (MCC)) existentes en el área de cobertura del dispositivo móvil.
1300. En versiones de Google Maps anteriores a 2017, se requería disponer de una conexión de datos (telefonía móvil o Wi-Fi) para funcionar y acceder a la información cartográfica de los mapas. Sin embargo, una de las capacidades que se añadieron a la aplicación Maps fue la de "zonas sin conexión", que posibilita la navegación aunque no haya conexión de datos activa si se ha descargado previamente el mapa correspondiente a esa zona. Todos los detalles sobre el uso y configuración de las zonas sin conexión pueden consultarse en [Ref.- 227].
1301. A partir de la versión 9.12 de Google Maps para Android, se incorporó la opción de menú "Tu cronología" [Ref.- 228].
1302. Para crear la cronología, además de la ubicación, el historial de ubicaciones debe estar activo en "Ajustes - [Personal] Ubicación - [Servicios de ubicación] Historial de ubicaciones de Google" (consultar apartado 12.5. Historial de ubicaciones). Dado que esta opción se habilita para la cuenta de Google del usuario, el activarla implica que Google recordará la ubicación en todos los dispositivos en los que se inicie sesión con esa cuenta de Google. No obstante, se puede desactivar la opción correspondiente al dispositivo móvil si no se desea que los datos que se recopilen de él pasen a formar parte del historial de ubicaciones:



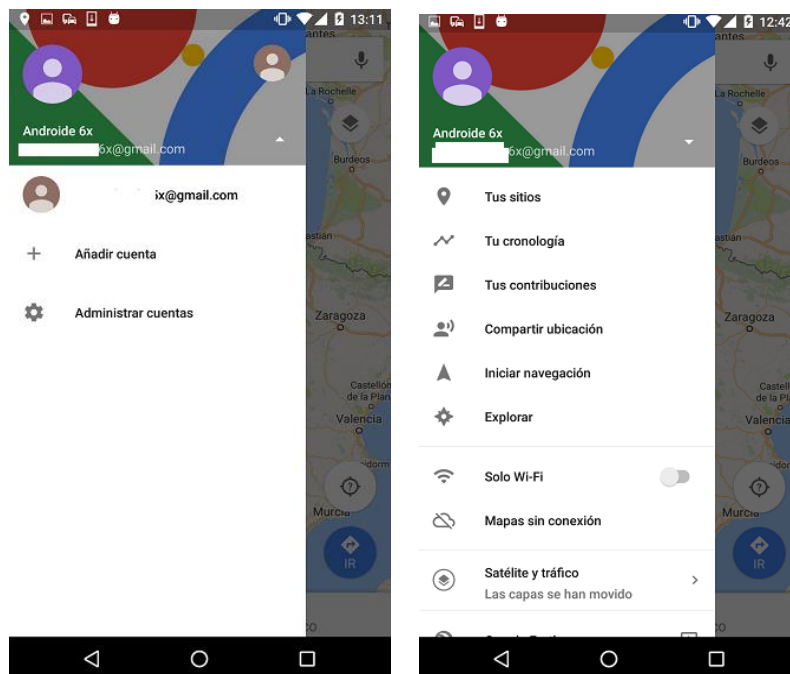
1303. Se enviarán a Google datos de ubicación de los dispositivos que tengan activa la opción del historial de ubicaciones, aunque no se esté utilizando ningún producto de Google.
1304. Si los servicios de ubicación están activos, el dispositivo móvil compartirá la ubicación con las aplicaciones que tengan permiso para ello.
1305. Con el cambio en el modelo de permisos de Android, la aplicación Maps dejó de tener concedidos por defecto permisos muy sensibles, como "inhabilitar el bloqueo de pantalla", "añadir o eliminar cuentas", "editar o borrar el contenido de USB". Para verificar qué permisos tiene concedidos en un momento dado en el dispositivo móvil, se recurrirá a "Ajustes - [Dispositivo] Aplicaciones - Maps - Permisos". Para evitar potenciales vulnerabilidades de seguridad, se recomienda mantener la app actualizada y conceder únicamente el permiso de ubicación, que es el único que requiere para su correcto funcionamiento:



1306. Hay que resaltar que estos permisos hacen referencia a permisos de sistema, no a las capacidades que tiene la aplicación Maps para interactuar con otros servicios de Google (como Google Fotos) y que afectan directamente a la privacidad del usuario. Por ejemplo, la opción "Correos electrónicos de la cronología" disponible en "Maps - Ajustes - Contenido personal", marcada por defecto, hará que Maps envíe periódicamente información al e-mail del usuario relacionada con los sitios visitados:



1307. Si el dispositivo móvil tiene varias cuentas de usuario de Google dadas de alta, es posible seleccionar qué cuenta se utilizará para la actividad registrada por la app Maps. Para cambiar entre cuentas (o añadir una nueva), se deben abrir los ajustes de Maps y seleccionar la cuenta a emplear (en versiones anteriores de Google Maps, era preciso marcar la opción "Cerrar sesión en Google Maps" para poder acceder a la configuración de otra cuenta). Desde el menú desplegable de las cuentas dadas de alta en Maps es posible también añadir cuentas adicionales:

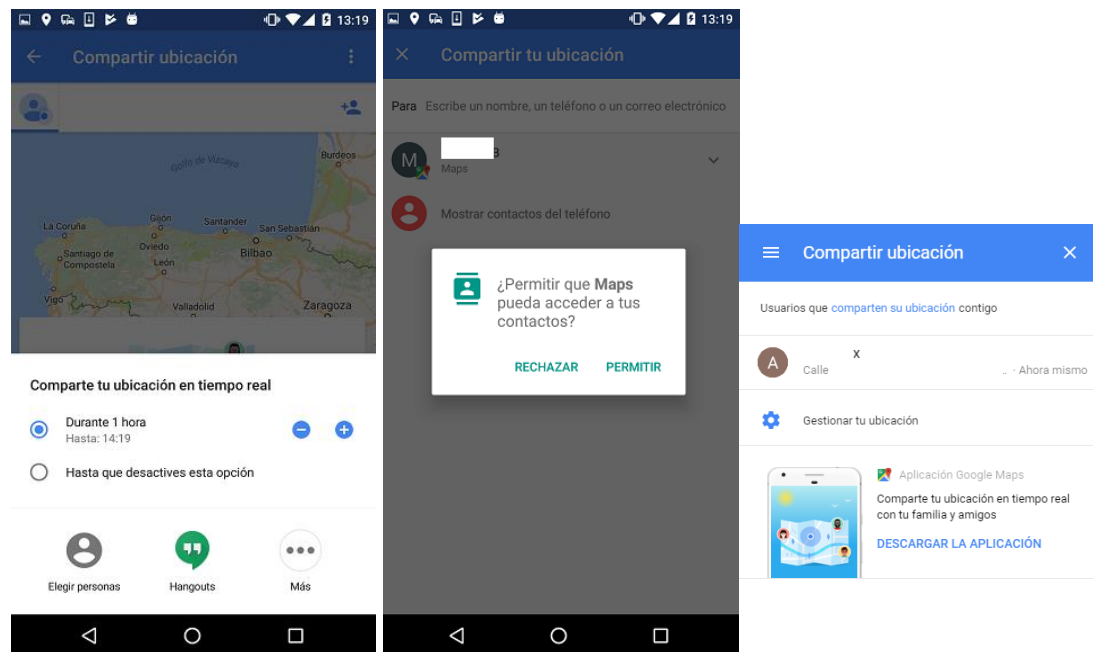


1308. La cuenta de usuario que está siendo utilizada por la app Maps es la que se muestra en la parte superior del menú de "Ajustes de Maps", que figura tras el icono de un rostro.

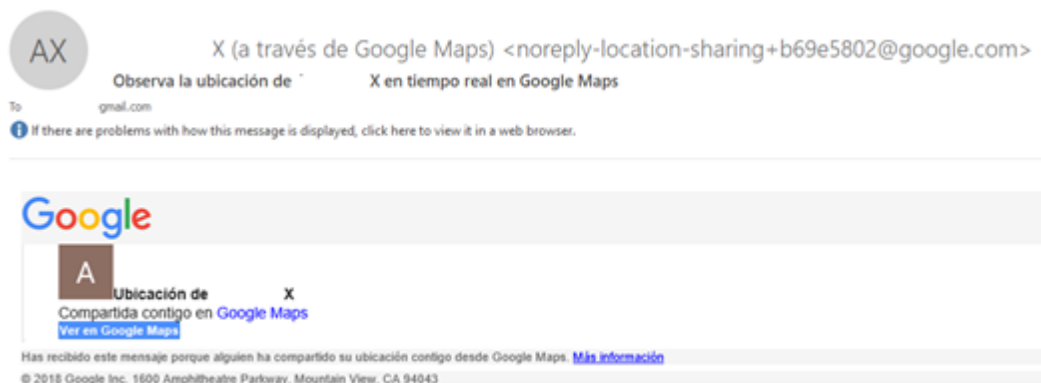
1309. Dada la sensibilidad de los datos que pueden recopilarse mediante Google Maps, y para preservar la privacidad del usuario, se recomienda deshabilitar el historial de ubicaciones y eliminarlo si ha estado activo en algún momento. Para ello, se puede acceder al menú "Ajustes [Personal] - Ubicación [Servicios de ubicación] Historial de ubicaciones de Google" o emplear el menú "Ajustes de Maps - Ajustes de ubicación de Google".
1310. Adicionalmente, si durante el uso de Maps se presenta el siguiente mensaje, se recomienda pulsar en "Omitir":



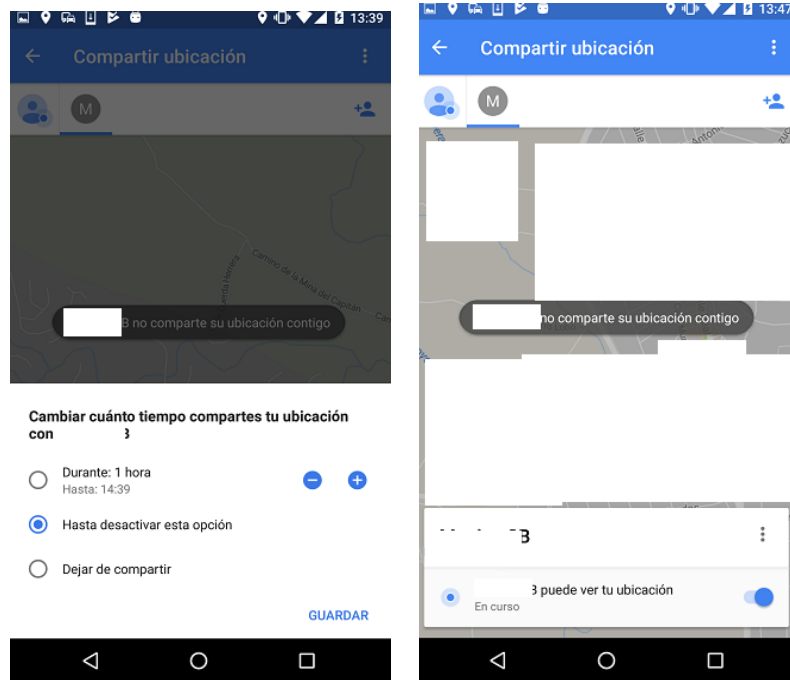
1311. Una opción que se añadió a la app Maps en las versiones posteriores a marzo del 2017 fue la de "Compartir ubicación" [Ref.- 230], que permite que el usuario comparta su posición geográfica con los contactos de su elección.
1312. Dicha opción permite conceder permisos para ver la ubicación en tiempo real del dispositivo durante el tiempo especificado por el usuario. Esta opción se habilita a través del menú de la app Maps denominado "Compartir ubicación".
1313. Durante el proceso de compartición de ubicación, Maps solicitará al usuario acceso a sus contactos en el teléfono. Es importante reseñar que, aunque este permiso no se conceda, y que, por tanto, los contactos no se puedan recopilar de los contactos del teléfono, Maps sí tendrá acceso a los contactos que el usuario haya añadido a su cuenta de Google. Por tanto, el denegar acceso a los contactos no implica que la app Maps pueda consultarlos a través de la cuenta de Google activa:



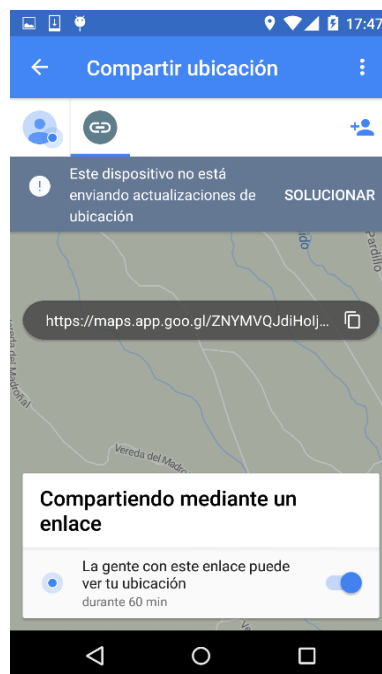
1314. Si se desea compartir la ubicación con un contacto, éste ha de disponer de una cuenta de usuario de Google. En ese caso, el usuario con el que se ha compartido la ubicación recibirá un mensaje de correo en su cuenta de Gmail que le informa de que otra persona ha compartido su ubicación con él:



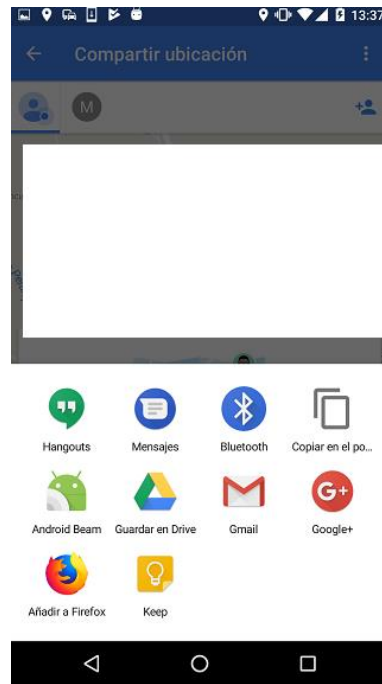
1315. En versiones anteriores de Maps, la compartición de la ubicación con otros usuarios tenía una duración de 1 hora (por defecto) y de un máximo de 72 horas. En la versión empleada para la elaboración de la presente guía, no hay un máximo de tiempo definido, sino un interruptor que desactiva la compartición:



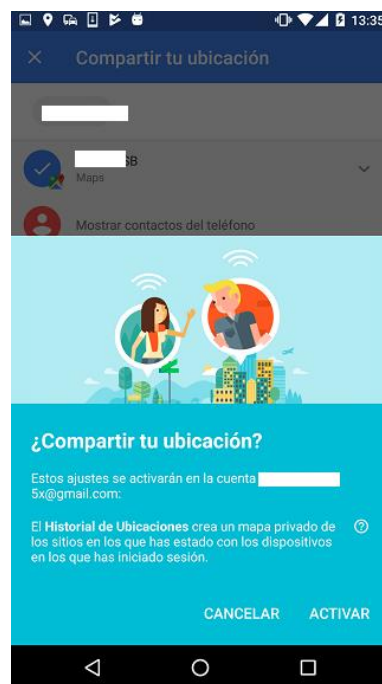
1316. Si el contacto no dispone de una cuenta de usuario de Google o no se desea hacer uso de ella para acceder a la ubicación del dispositivo, se ha de crear un enlace mediante el menú de la app Maps, "Compartir ubicación - Más - Copiar en el portapapeles". Seguidamente, se puede compartir el enlace creado mediante la aplicación o servicio que se desee, y así se dará acceso a la ubicación durante un máximo de 1 hora.



1317. Es posible también elegir una aplicación de mensajería instalada en el dispositivo móvil para compartir la ubicación seleccionando la misma a través del menú de la app Maps, "Compartir ubicación - Más - [seleccionar la aplicación de la lista mostrada]".



1318. Para poder compartir la ubicación, es necesario habilitar el historial de ubicaciones, tal como se informa en el mensaje que aparece al intentar compartir la ubicación.



1319. Las opciones de uso de la opción "Compartir ubicación" están disponibles en la ayuda oficial de Google Maps en tiempo real [Ref.- 231].
1320. Por cuestiones de privacidad, se recomienda no hacer uso de esta funcionalidad. En caso de requerirse por alguna circunstancia, se debe verificar que el canal de compartición es seguro y limitar el acceso en tiempo a la misma todo lo posible.

1321. Por cuestiones de privacidad, no se recomienda en absoluto dar de alta la dirección de casa o del trabajo en el área de "Ajustes de Mapa - Ajustes - Editar casa o trabajo".
1322. El resto de opciones de uso de la app Maps para navegación quedan fuera del alcance de la presente guía, pero se puede obtener información detallada al respecto en la "Ayuda de Google Maps" [Ref.- 229].

12.3. A-GPS

1323. *Assisted GPS* (A-GPS o aGPS) es un sistema de ayuda para mejorar la obtención de información y la precisión de los datos de localización obtenidos mediante GPS, que permite disponer de una estimación de la ubicación del dispositivo móvil más rápidamente.
1324. Especialmente A-GPS permite agilizar significativamente el tiempo necesario para ubicar inicialmente el dispositivo móvil, parámetro conocido como TTFF (Time-To-First-Fix).
1325. El módulo A-GPS está (teóricamente) disponible en dispositivos móviles Android con GPS, especialmente cuando se hace uso del modo de alta precisión.
1326. El sistema A-GPS es un sistema híbrido que se basa en obtener información de las torres de telefonía móvil detectadas desde la ubicación del dispositivo móvil (y que proporcionan información auxiliar o de asistencia sobre la ubicación), y adicionalmente, hace uso de las conexiones de datos (habitualmente 2/3/4G, o Wi-Fi) del dispositivo móvil para obtener información actualizada de dónde se encuentran los diferentes satélites GPS en un momento temporal (información con datos de las órbitas y de sincronización - reloj - de los satélites, conocido también como almanaque).
1327. Por tanto, la funcionalidad A-GPS también es visible a través del tráfico de datos del dispositivo móvil. Android 7.x, en diferentes momentos temporales como por ejemplo durante el proceso de arranque del dispositivo móvil, envía tráfico no cifrado hacia los servidores de gpsOneXTRA (ver apartado "12.4. gpsOneXTRA").
1328. Debe tenerse en cuenta que, pese a que el uso del módulo GPS está siempre habilitado por defecto si los servicios de ubicación están activos, salvo que se haga uso del modo de ahorro de batería, hay escenarios en los que no es posible obtener señal de GPS (por ejemplo, en el interior de edificios), por lo que las apps como Maps harán uso automáticamente de las capacidades de ubicación sin hacer uso del módulo GPS si estas han sido habilitadas, por ejemplo, en el modo de alta precisión.
1329. En resumen, se recomienda deshabilitar la funcionalidad A-GPS mediante la activación del modo "Solo en dispositivo" salvo que se quiera hacer uso explícito de estas capacidades, con el objetivo de evitar desvelar información de la ubicación del dispositivo móvil (por ejemplo, torres de telefonía móvil o redes Wi-Fi cercanas) a través del tráfico de datos cifrado destinado a los servidores de Google.

12.4. GPSONEXTRA

1330. Con el objetivo de agilizar el proceso de localización, complementando las coordenadas obtenidas del módulo GPS y su precisión, los dispositivos móviles Android pueden disponer de la funcionalidad asociada a la asistencia mediante gpsOneXtra (aplicación conocida como QuickGPS en otras plataformas móviles).
1331. Hasta Android 5.x inclusive, esta funcionalidad se conectaba a un servidor web del dominio "gpsonextra.net" (xtra1, xtra2 ó xtra3) para descargar la información semanal de la localización de los satélites GPS alrededor del planeta (o almanaque). Por defecto, la descarga automática de esta información está habilitada en Android. A partir de Android 6.x, la conexión se realiza con servidores web del dominio "izatcloud.net" (en el ejemplo utilizado para la presente guía, concretamente con el servidor "xtrapath1").
1332. La descarga se lleva a cabo mediante los servidores de "izatcloud.net" mencionados previamente, descargándose el fichero "/xtra3grc.bin".

Nota: La configuración interna de Android tanto del servidor de tiempos NTP como de los servidores de gpsOneXTRA y A-GPS (ver apartado "12.3. A-GPS") está disponible en el fichero "/system/etc/gps.conf", aunque no se dispone de ningún ajuste de configuración a través del interfaz gráfico de usuario de Android para modificar estos parámetros. Ejemplo (parcial) del fichero "gps.conf":

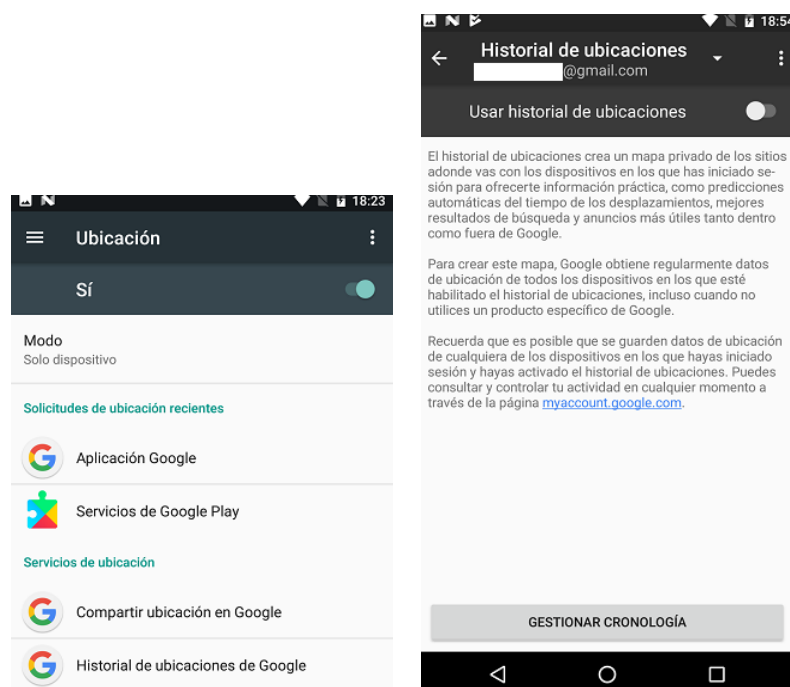
```
XTRA_SERVER_1=http://xtrapath1.izatcloud.net/xtra3grc.bin
XTRA_SERVER_2=http://xtrapath2.izatcloud.net/xtra3grc.bin
XTRA_SERVER_3=http://xtrapath3.izatcloud.net/xtra3grc.bin
```

1333. Las conexiones gpsOneXTRA ("xtrapathN.izatcloud.net", TCP/80), para incrementar la precisión de localización del GPS, se llevan a cabo (por ejemplo) cada vez que se enciende el dispositivo móvil y éste dispone de acceso a Internet.
1334. A finales del año 2016 se identificó una vulnerabilidad de denegación de servicio que afectaba a los dispositivos móviles Android y que podía ser explotada mediante la manipulación de los ficheros de posicionamiento obtenidos mediante el servicio gpsOneXtra [Ref.- 223].

12.5. HISTORIAL DE UBICACIONES

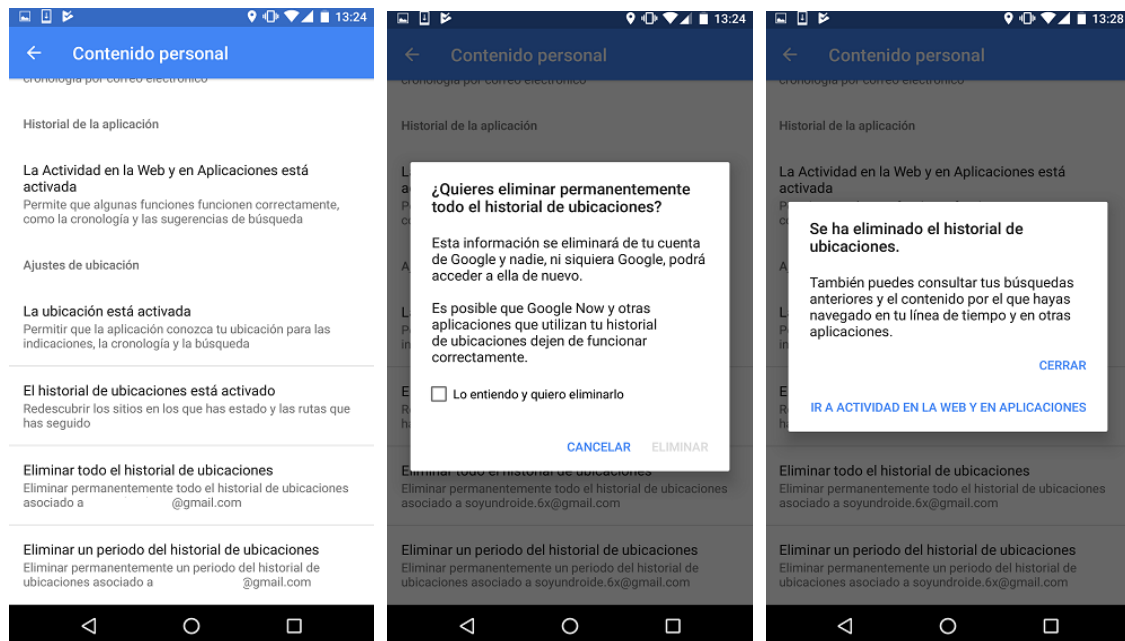
1335. Los servicios de Google pueden hacer uso de los datos del historial de ubicaciones para, por ejemplo, mejorar los resultados de búsqueda de Google Maps (mediante la app Maps) en función de los sitios frecuentados habitualmente por el usuario [Ref.- 167].
1336. El historial de ubicaciones almacena un registro histórico de la ubicación del usuario procedente de todos los dispositivos móviles en los que el usuario haya iniciado sesión con su cuenta de usuario de Google. Para que se envíe la información al historial, y se pueda almacenar, el dispositivo móvil debe tener habilitado el historial de ubicaciones.

1337. El uso de la opción "historial de ubicaciones" permite a Google utilizar (y potencialmente almacenar) periódicamente los datos de ubicación más recientes del dispositivo móvil, así como la información sobre las actividades que conllevan movimiento (conducir, caminar, pasear en bicicleta, etc.) y asociarlos a la cuenta de usuario de Google, incluso aunque no se esté utilizando ningún producto o servicio de Google.
1338. La frecuencia con la que los informes de ubicación actualizan los datos de ubicación no es fija y depende de diferentes factores, como la batería del dispositivo móvil, si el usuario se está desplazando, o la velocidad a la que se desplaza.
1339. A través del menú "Ajustes [Personal] - Ubicación", y de la sección "Servicios de Ubicación", es posible seleccionar "Historial de ubicaciones de Google" y acceder a los ajustes de configuración para activar o desactivar el historial de ubicaciones de la cuenta de usuario de Google, tanto a nivel global como a nivel de dispositivo:

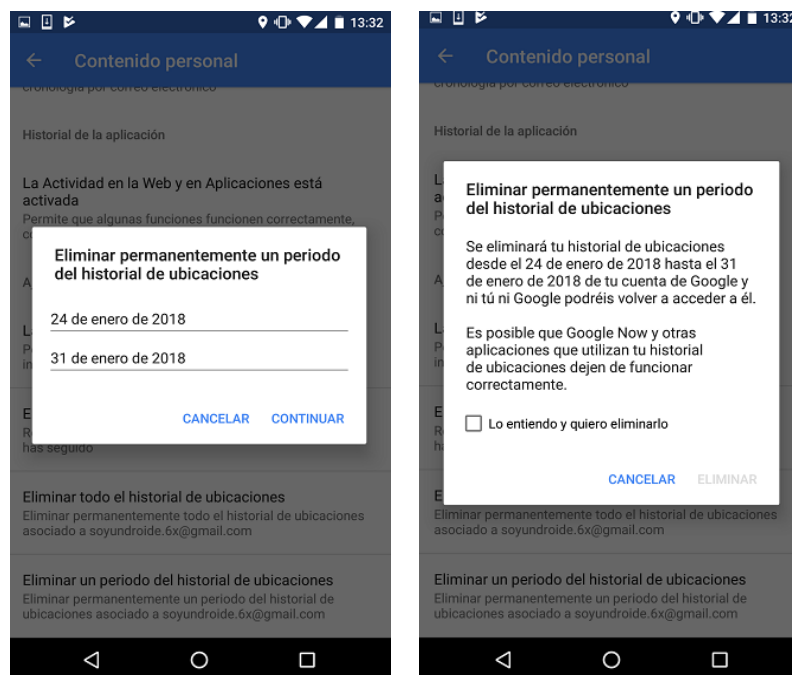


1340. La configuración del historial de ubicaciones está disponible a través del menú mencionado previamente, pero, aunque se desactive a través de las opciones descritas, no se eliminarán los datos recopilados hasta ese momento.
1341. Si se desea eliminar el histórico, es necesario acceder a "Ajustes - [Personal] Ubicación - [Servicios de Ubicación] Historial de ubicaciones de Google - <...> Ver/Administrar", o, alternatively, pulsar en el botón "Gestionar Cronología". Esto hará que se lance la app Maps (app empleada por defecto) en una ventana que muestra la ubicación actual y las demás ubicaciones registradas para el día actual; desde el menú "<...>" accesible desde la esquina superior derecha de la pantalla, se puede seleccionar "Ajustes - [Contenido personal] - Eliminar todo el historial de ubicaciones".

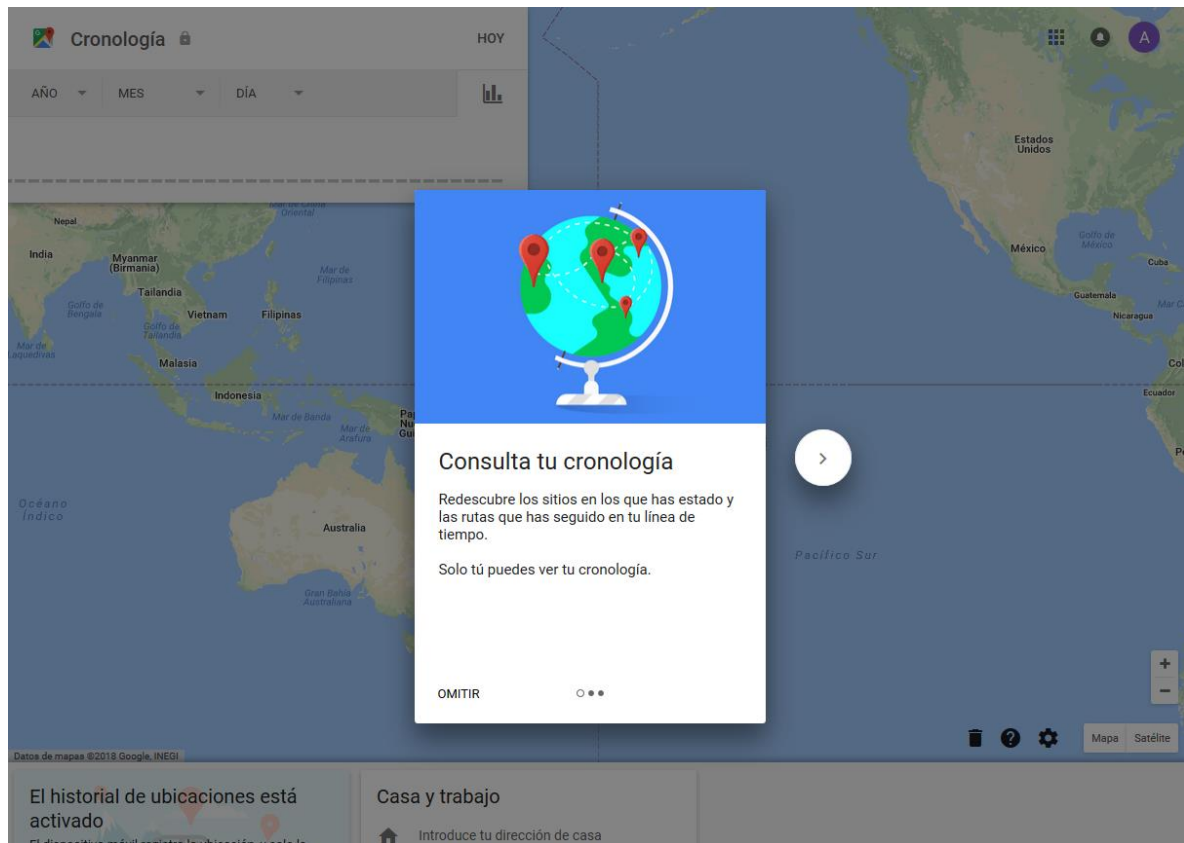
Tras aceptar el mensaje de confirmación, se borrarán los datos de ubicaciones previamente almacenados por Google:



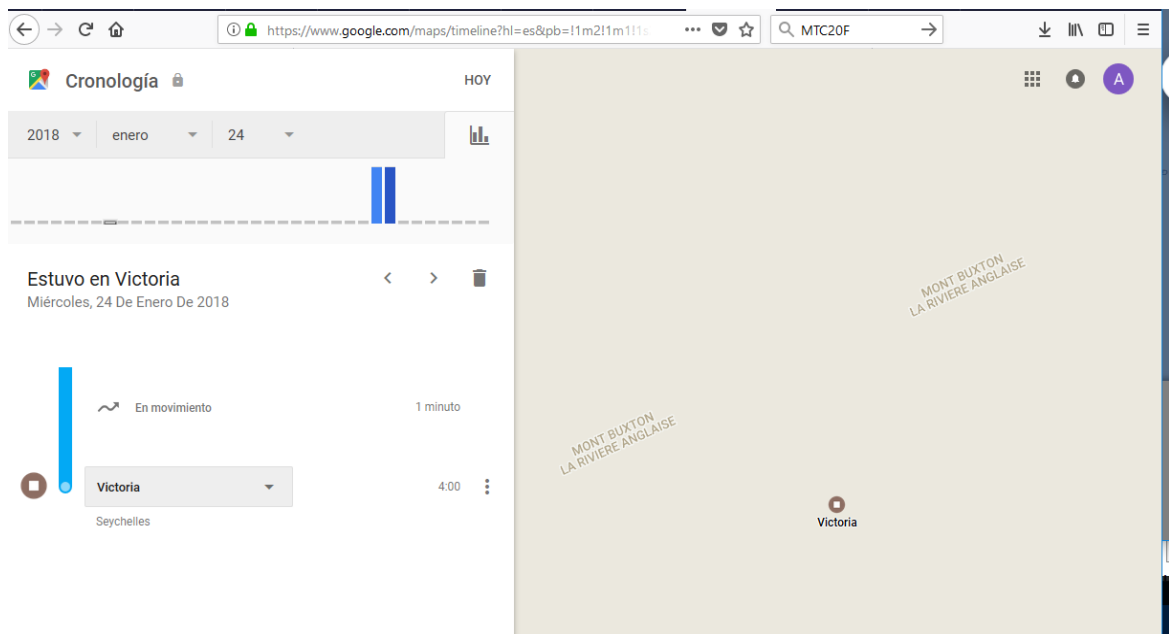
1342. También es posible eliminar un periodo concreto del historial, seleccionando la fecha de inicio y de fin.



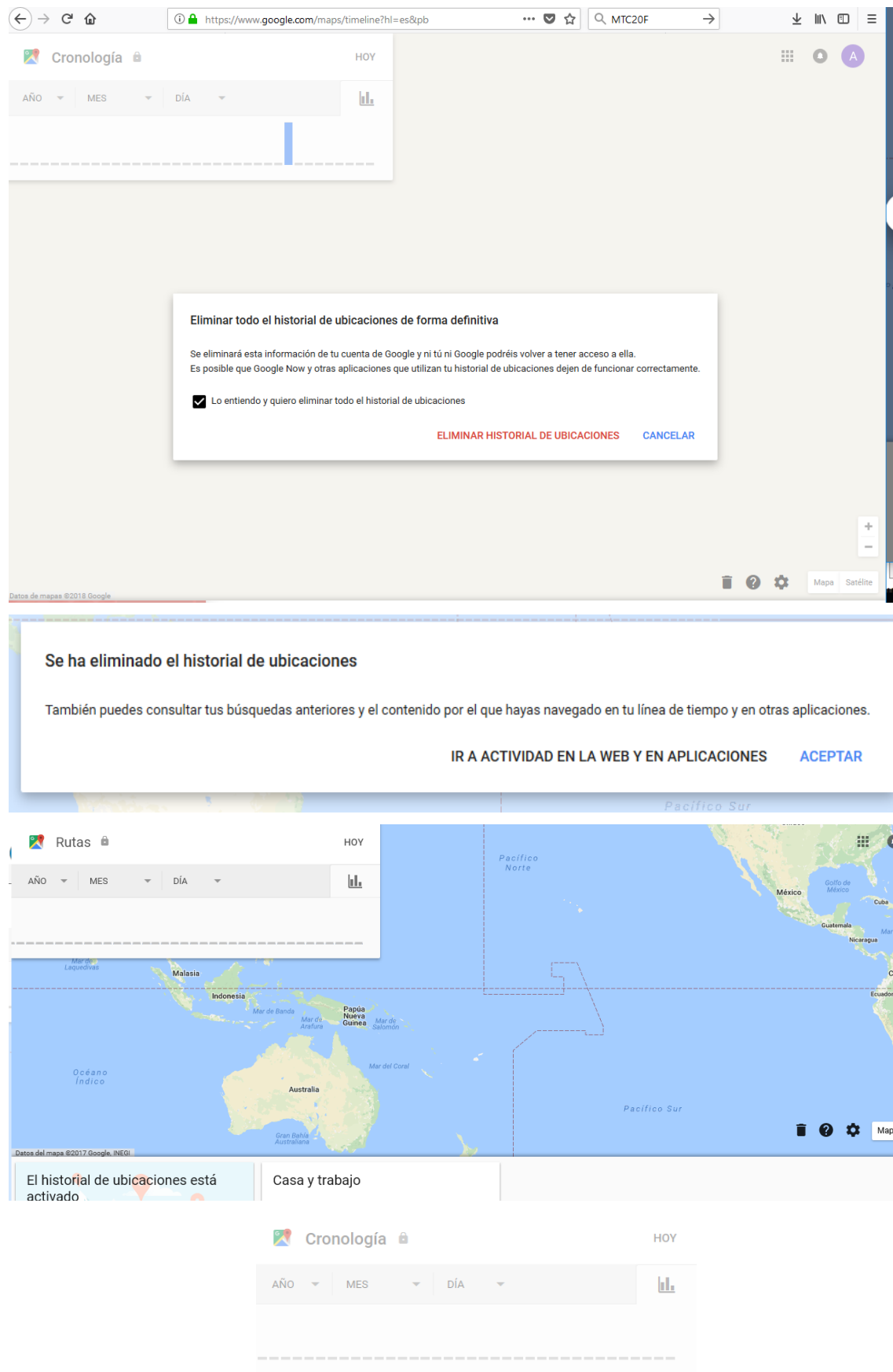
1343. Finalmente, un usuario puede también acceder y administrar el historial de ubicaciones vinculado a su cuenta de usuario de Google a través de la siguiente URL: "https://maps.google.com/locationhistory/". Esta funcionalidad está asociada al "Historial de la cuenta", disponible desde "https://myaccount.google.com/activitycontrols" (y que también permite el acceso al historial de actividad web y de aplicaciones).



1344. Desde la misma es posible obtener los detalles de las ubicaciones registradas para el usuario, incluso clasificadas por fechas y horas, y obtener información de distancias entre ubicaciones:



1345. Desde las opciones situadas en el margen inferior derecho de la página web, es posible eliminar los registros del historial y pausar la recolección de datos. Cuando el historial ha sido eliminado, desde el dispositivo móvil o desde la página web, no se dispone de datos asociados a esta funcionalidad:



1346. En resumen, desde el punto de vista de seguridad, y, especialmente, de la privacidad el usuario, no se recomienda hacer uso del historial de ubicaciones (es decir, permitir a Google que almacene un histórico de todos

los datos de ubicación). Adicionalmente, se recomienda borrar el historial de ubicaciones actualmente existente en la cuenta de usuario de Google.

12.6. OTRAS APPS QUE HACEN USO DE LA LOCALIZACIÓN

Nota: Queda fuera del alcance de la presente guía realizar un análisis detallado de seguridad de las diferentes apps que hacen uso de los servicios de localización de Google en Android, así como de la existencia de posibles vulnerabilidades en las mismas.

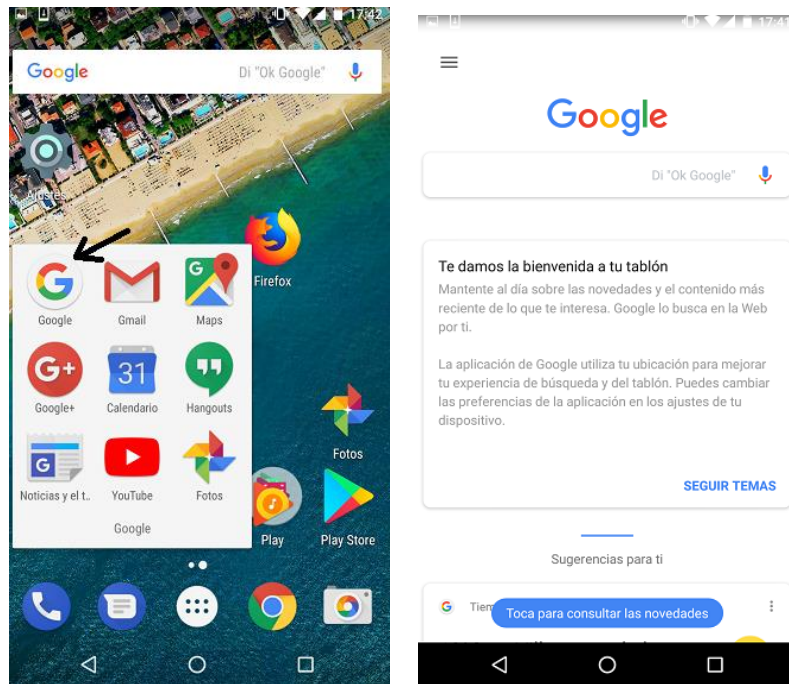
A continuación, se muestran ejemplos de algunas apps existentes por defecto que hacen uso de los servicios de localización. Únicamente se recomienda verificar que las mismas hagan uso de tráfico cifrado mediante HTTPS, y si exponen o no los datos intercambiados entre ellas y los servidores remotos mediante tráfico no cifrado, normalmente empleando el protocolo HTTP. A modo de ejemplo, en algunos casos el valor del agente de usuario (o "User-Agent") desvelado por las apps proporciona numerosos detalles sobre la app y el dispositivo móvil asociado.

Se recomienda no hacer uso de apps o servicios que desvelan información detallada de la ubicación del usuario y del dispositivo móvil empleado, y que hacen uso de conexiones HTTP no cifradas, salvo que explícitamente se desee acceder a la información que éstos proporcionan.

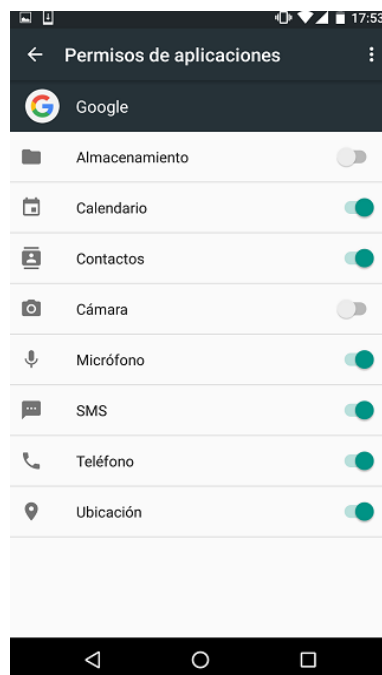
La mayoría de apps mencionadas requieren de una conexión de datos (telefonía móvil o Wi-Fi) para funcionar y acceder a la información necesaria para su funcionalidad.

12.6.1. BUSCADOR DE GOOGLE

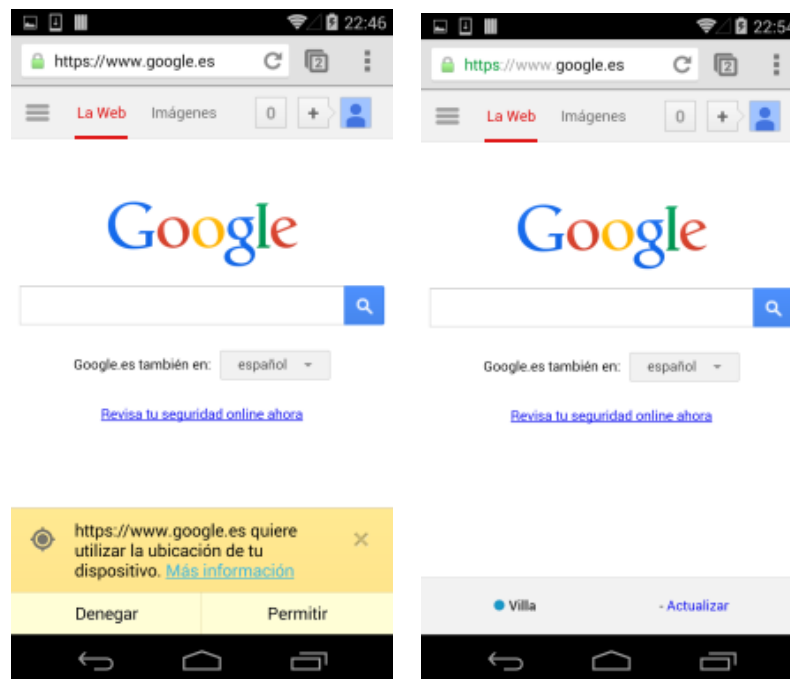
1347. La app "Google", que proporciona acceso directo al buscador de Google y a otros servicios de la compañía, puede hacer uso también de los servicios de localización, integrándolos directamente en sus búsquedas y en las capacidades de notificaciones personalizadas asociadas a la actividad del usuario.
1348. El buscador de Google está integrado en la aplicación "Google", y se accede a él pulsando en la barra de búsqueda situada bajo el banner "Google":



1349. Android concede a la app Google durante su instalación multitud de permisos, entre los que se encuentra el de ubicación. Se recomienda deshabilitarlo salvo que se requiera por alguna causa:



1350. En versiones anteriores de Android, el acceso al buscador de Google directamente desde el navegador web de Android (descrito posteriormente en el apartado "12.6.3. Navegador web"), solicitaba al usuario utilizar la ubicación del dispositivo móvil, pudiendo éste permitir o denegar dicho acceso (dicho comportamiento se ilustra en las imágenes siguientes). Sin embargo, desde Android 6.x y las versiones más recientes del navegador web y de la app Google, el permiso de ubicación está concedido por defecto para ambas, por lo que ya no lo solicitan explícitamente al usuario.



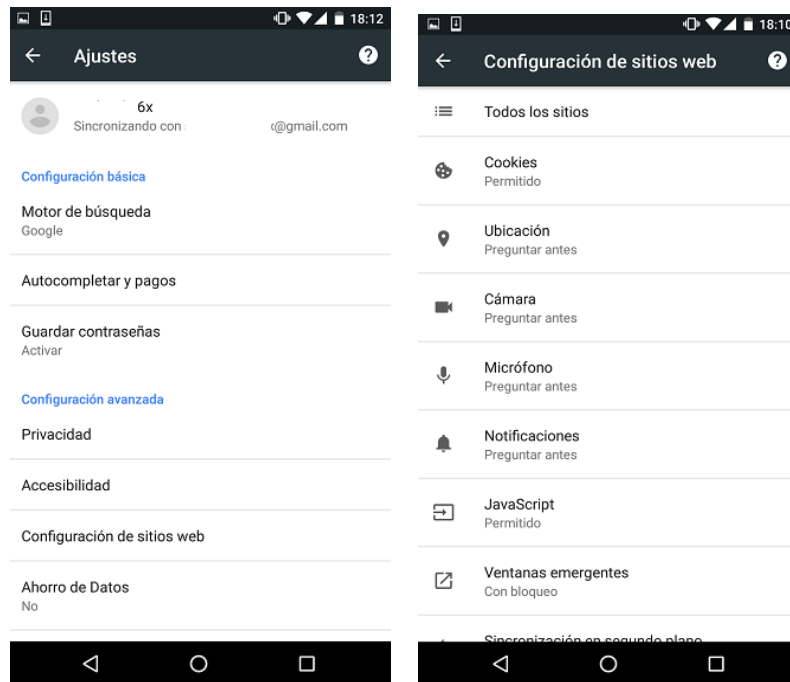
1351. Los permisos habilitados anteriormente para ciertos sitios web pueden ser modificados posteriormente (ver apartado "12.6.3. Navegador web").

12.6.2. REDES SOCIALES

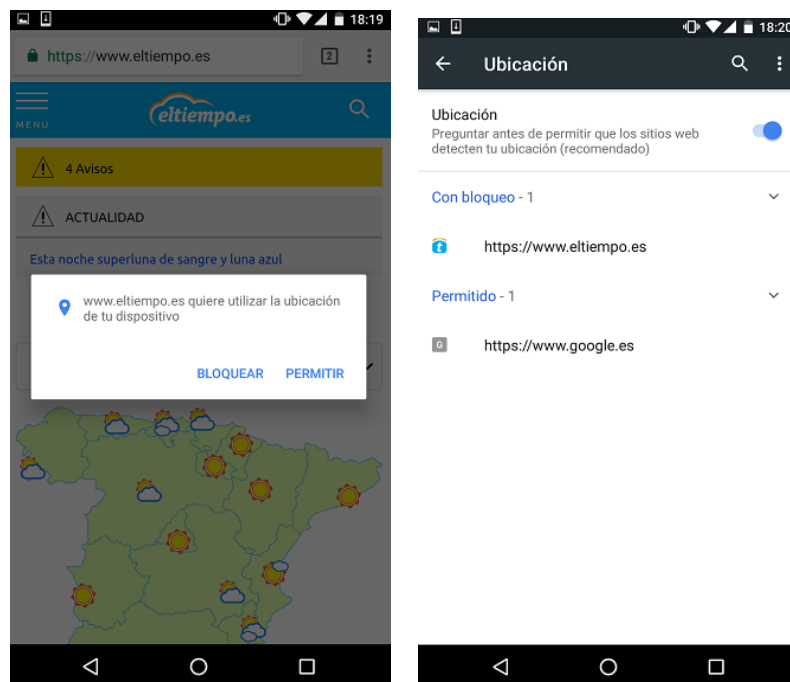
1352. La información de localización puede integrarse también con las apps de acceso a redes sociales, siendo posible la publicación de la ubicación del usuario a través, por ejemplo, de Twitter o Facebook.
1353. Debido a que las apps asociadas a estas redes sociales no están disponibles por defecto en Android 7.x (sí lo estaban en versiones previas de Android), no se profundizará en los detalles centrados en deshabilitar el uso de los servicios de localización desde las mismas.
1354. Desde el punto de vista de seguridad, se recomienda no hacer pública la ubicación en la que se encuentra el usuario en redes sociales o servicios avanzados similares con el objetivo de proteger su privacidad.

12.6.3. NAVEGADOR WEB

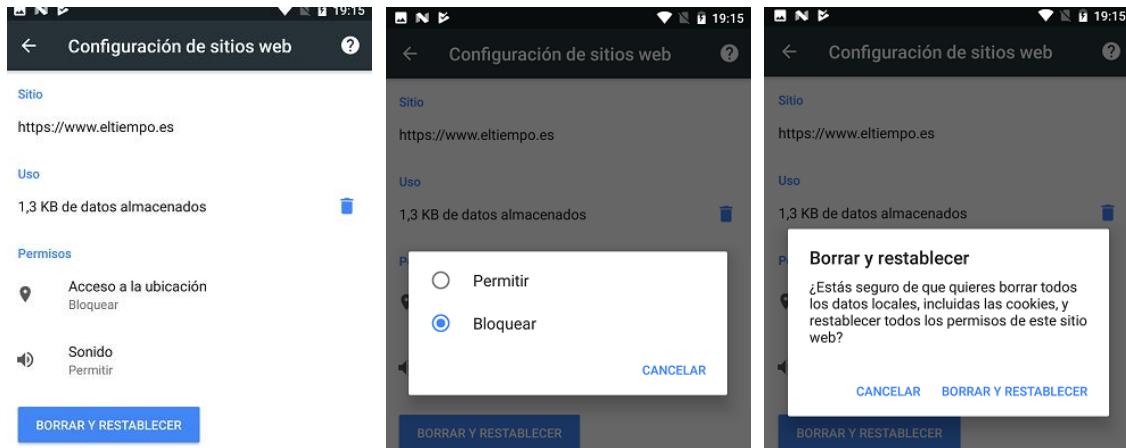
1355. El navegador web existente por defecto en Android, Chrome, permite autorizar el acceso a la información de localización (o ubicación del dispositivo móvil) o bloquearlo de manera general. Para ello, se ha de acceder al menú de "Ajustes" de "Chrome", y en concreto a través de la opción "Ajustes - [Configuración] Configuración de sitios web - Ubicación". La opción asociada permite que se solicite al usuario autorización cuando una página web trate de detectar la ubicación del usuario o, si se desactiva, bloquear cualquier solicitud de ubicación por parte de cualquier página web. Esta opción sólo entrará en juego si Chrome tiene concedido el permiso de ubicación a nivel de aplicación en los ajustes de configuración de Android.



1356. Si Chrome tiene permiso de ubicación y la opción "Ubicación - Preguntar antes" está activa, tan pronto se visite una página web que solicite acceso a la ubicación del usuario, se creará bajo el menú anterior la lista "Permitido" (si se concedió acceso) o la lista "Con bloqueo" (si se denegó el acceso).



1357. Los permisos de acceso a la ubicación habilitados anteriormente para ciertos sitios web pueden ser modificados posteriormente desde el menú de "Ajustes" de "Chrome", y en concreto a través de la opción "Ajustes - [Opciones avanzadas] Configuración de sitios web - Ubicación". Android dispone de capacidades granulares para alterar el acceso a la información de localización para cada sitio web. Tras seleccionar un sitio web concreto, se puede cambiar su permiso "Acceso a la ubicación" seleccionando el nombre del sitio y eligiendo entre las opciones "Permitir" o "Bloquear":



1358. El botón "Borrar y restablecer" provocará que se elimine la configuración de permisos asociada al sitio web, además de borrar sus cookies.
1359. Desde el punto de vista de seguridad se recomienda deshabilitar la opción "Ubicación" para todos los sitios web, con el objetivo de no permitir a ningún sitio web disponer de acceso a la información de localización, incluso aunque tuvieran este acceso previamente permitido de manera individual.

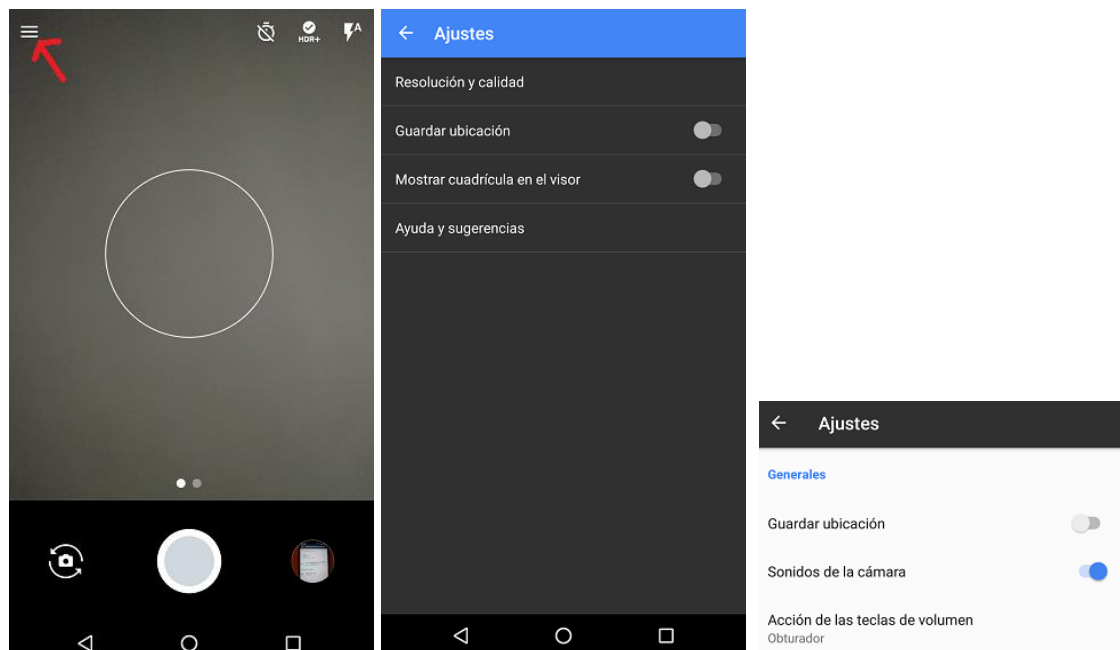
12.7. INFORMACIÓN GEOGRÁFICA EN FOTOGRAFÍAS

1360. Android permite añadir información de la localización geográfica a las fotografías y vídeos realizados con la cámara del dispositivo móvil. Al realizar la fotografía (o vídeo) se añaden las coordenadas GPS en el momento de tomar la instantánea a la cabecera EXIF de la imagen.
1361. En función de la versión de la app "Cámara", es posible se pregunte al usuario si desea etiquetar las fotografías y vídeos con la información de la ubicación dónde se han realizado (opción habilitada por defecto), información también accesible para otras apps:



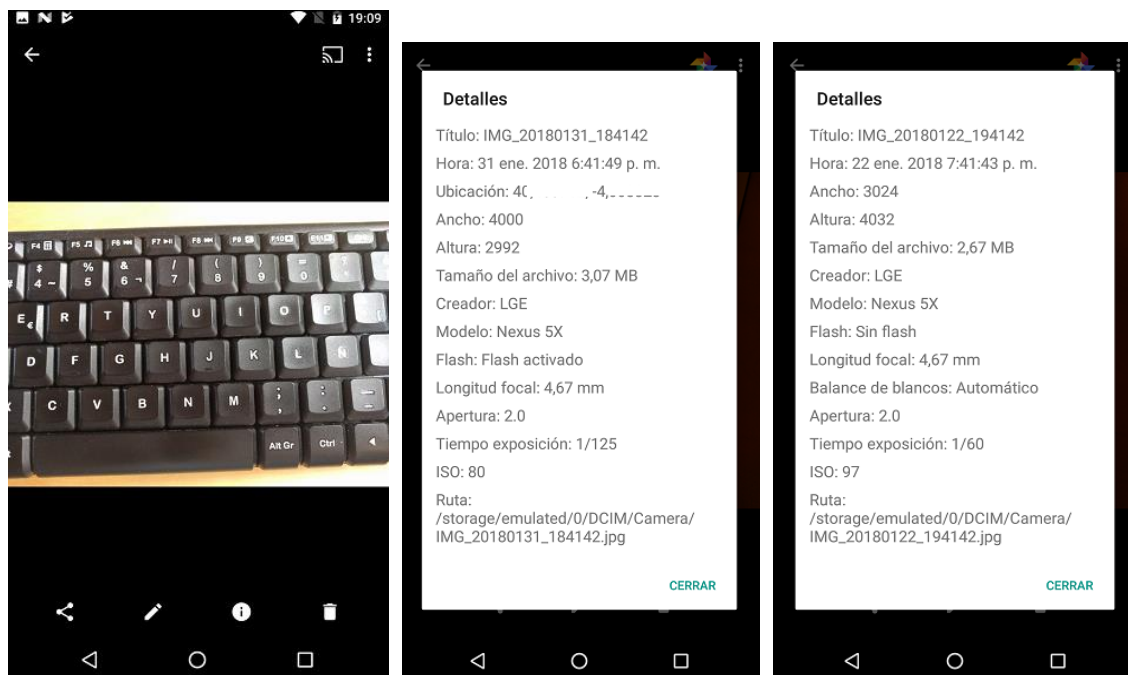
1362. Desde el punto de vista de seguridad se recomienda deshabilitar esta funcionalidad con el objetivo de no desvelar detalles de las ubicaciones del usuario.

1363. Esta funcionalidad puede ser habilitada y deshabilitada en cualquier momento a través del menú de configuración de la cámara. Para ello es necesario abrir la cámara mediante la app "Cámara" desde el *Launcher*, pinchar en el menú de ajustes señalado con una flecha en la siguiente imagen y seleccionar "Ajustes":

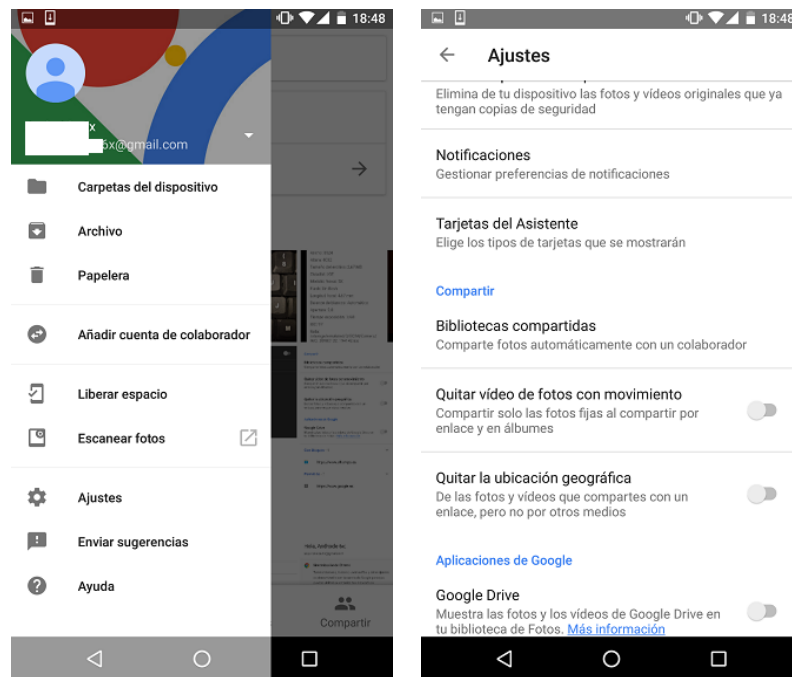


1364. La opción "Guardar ubicación" permite habilitar o deshabilitar esta funcionalidad. En caso de elegir su almacenamiento, la app "Cámara" guarda las coordenadas GPS que tenía el dispositivo en el momento de tomarse la fotografía.

1365. Los detalles de ubicación de las fotografías pueden ser visualizados desde el dispositivo móvil Android, y en concreto, desde la galería de imágenes de la cámara. La galería está disponible tanto desde la app "Cámara", desplazando desde la derecha hacia la izquierda la imagen de la cámara, o también desde la app "Fotos" (en versiones anteriores de Android, existía también la app "Galería", pero desde Android 6.x ya no está disponible, y la funcionalidad que ésta tenía asociada se integra dentro de la app "Fotos").
1366. Para ello es necesario emplear el botón de menú, y la opción "(i)" (ejemplo asociado a la app "Fotos"):



1367. El campo "Ubicación" (al comienzo de la lista de atributos de la imagen) indicará la ubicación dónde fue tomada (latitud y longitud). En caso de no disponer de la información de ubicación de la imagen, este campo o atributo no será mostrado.
1368. Respecto a la app "Fotos", dentro de sus ajustes incluye una opción ("Quitar la ubicación geográfica") que permite que, aunque una imagen o un vídeo hayan sido tomados con el permiso de ubicación activo para la cámara, al compartir dicho elemento a través de un enlace, se elimine la información de ubicación. Se recomienda activar dicha opción para evitar que la información de ubicación pueda compartirse inadvertidamente a través de otra app.



1369. Aunque dentro de los ajustes de la app "Fotos" Google sugiere utilizar la ubicación para facilitar la organización y la búsqueda de imágenes, se recomienda no hacer uso de la funcionalidad que incluye la información del GPS en fotografías o vídeos, especialmente para su publicación o distribución en Internet, salvo que se quiera hacer uso explícito de estas capacidades. En caso contrario, las imágenes y vídeos revelarán los detalles exactos de donde han sido tomadas.

13. COMUNICACIONES USB

1370. La conexión USB del dispositivo móvil a un ordenador proporciona dos funcionalidades o tipos de conexiones diferentes: [Ref.- 224]
- Modo "accessory": el uso principal que se hace de este modo es para que el dispositivo móvil Android se comporte como una unidad de disco, compartiendo la tarjeta de almacenamiento externa (en caso de disponer de una) y otras capacidades de almacenamiento internas desde el dispositivo móvil al ordenador.
 - Los dispositivos móviles Android que no disponen de una tarjeta de almacenamiento externa física, simulan su existencia compartiendo parte de las capacidades de almacenamiento internas del dispositivo móvil.
 - Modo "host": un ejemplo de utilización de este modo es la posibilidad de ofrecer la conexión de telefonía móvil de datos, o *tethering*, para compartir la conexión de datos 2/3/4G o Wi-Fi del dispositivo móvil con el ordenador.
 - La compartición de la conexión a Internet a través de USB o Bluetooth puede hacer uso tanto de la conexión de datos 2/3/4G como de la conexión Wi-Fi del dispositivo móvil. La compartición a través de Wi-Fi (punto de acceso o zona Wi-Fi) sólo puede hacer uso

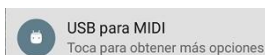
de la conexión de datos 2/3/4G, ya que el interfaz Wi-Fi no puede actuar simultáneamente de cliente y de punto de acceso.

1371. Ambas funcionalidades USB, "accessory" y "host", no pueden ser utilizadas simultáneamente. En caso de estar usando las capacidades de *tethering*, éstas deben ser deshabilitadas para tener acceso al almacenamiento vía USB, y viceversa.
1372. El modo de conexión por defecto a través del puerto USB del dispositivo móvil es el de sólo carga, comportamiento que fue introducido en Android 6.x [Ref.- 602], y no es posible definir otro modo que sea el que se emplee por defecto al establecer una conexión USB por primera vez.

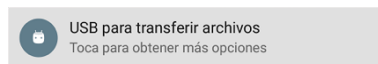
13.1. ACCESO AL ALMACENAMIENTO POR USB

Nota: El acceso al dispositivo móvil como unidad de almacenamiento USB puede estar disponible incluso con el modo de depuración USB habilitado en Android. En caso de conflictos, si está habilitado, basta con deshabilitar este modo (opción recomendada desde el punto de vista de seguridad), para disponer únicamente de las capacidades de almacenamiento USB estándar. El modo de depuración se puede deshabilitar mediante el menú "Ajustes [Sistema] - Opciones de desarrollo" (una vez habilitadas), desactivando la opción "Depuración USB".

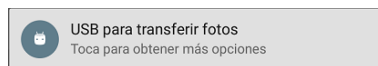
1373. Para acceder a las capacidades de almacenamiento USB basta con conectar el dispositivo móvil al ordenador mediante el puerto USB. Como resultado, se generará una notificación correspondiente en la barra superior de estado, "Cargando este dispositivo por USB", ya que, por defecto, ésta es la acción configurada tan pronto se detecte la conexión USB. Este tipo de conexión no involucra ninguna transferencia de datos entre el ordenador y el dispositivo móvil.
1374. Para habilitar otros usos del puerto USB (el de carga se mantiene en paralelo a otros que se puedan seleccionar), se abrirá la notificación y se elegirá el modo deseado en el menú "Utilizar USB para". Las opciones disponibles (además de "Cargar este dispositivo") son:
- "Suministrar corriente eléctrica": en caso de querer que el dispositivo móvil se comporte como una fuente de alimentación para el otro equipo conectado vía USB.
 - "MIDI": esta opción se introdujo con Android 6.x, y permite conectar al dispositivo móvil un dispositivo tipo MIDI (como un teclado) a controlar por una app específica (como un sintetizador).



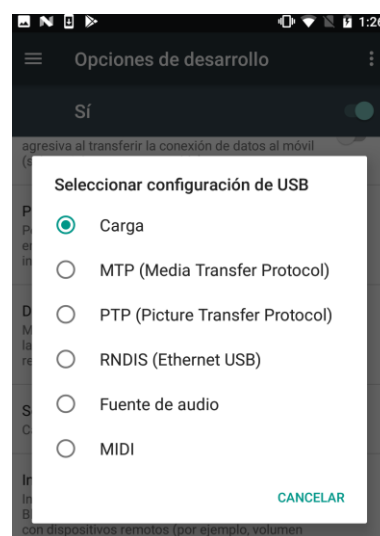
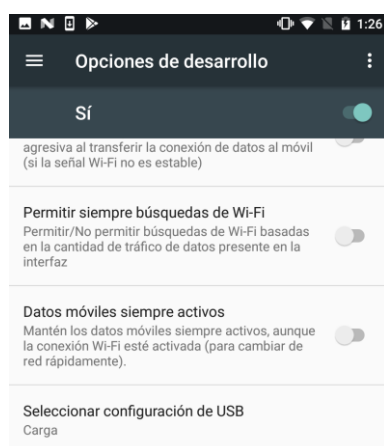
- "Transferir archivos": a través del protocolo MTP (Media Transfer Protocol), se permite que el dispositivo móvil emule a un dispositivo multimedia que proporciona acceso a distintos tipos de ficheros a través de múltiples carpetas, incluyendo las de la cámara (descritas a continuación). Este tipo de conexión permite el intercambio de ficheros entre el dispositivo móvil y el otro dispositivo en ambas direcciones.



- "Transferir fotos": a través del protocolo PTP (Photo Transfer Protocol), el dispositivo móvil emula una cámara de fotos (y/o vídeo) y proporcionará acceso únicamente a las fotos y vídeos desde dos carpetas, "DCIM" y "Pictures".



1375. También es posible seleccionar el modo de conexión del dispositivo a través del menú "Ajustes - [Sistema] Opciones de desarrollo - [Redes] Seleccionar configuración de USB":



1376. Para poder transferir datos, el usuario debe seleccionar otro modo de conexión a través de la notificación correspondiente (es decir, seleccionando la notificación que informa del tipo de conexión USB que está activa en ese momento), y elegir el tipo de transferencia que se desea ("Transferir archivos" o "Transferir fotos (PTP)"):

Utilizar USB para

- ☒ Cargar este dispositivo
Cargar solo este dispositivo
- ☐ Suministrar corriente eléctrica
Suministrar corriente eléctrica al otro dispositivo conectado
- ☐ Transferir archivos
Transferir archivos a otro dispositivo
- ☐ Transferir fotos (PTP)
Transferir fotos o archivos si no se admite MTP (PTP)
- ☐ Utilizar dispositivo como MIDI
Utilizar este dispositivo como MIDI

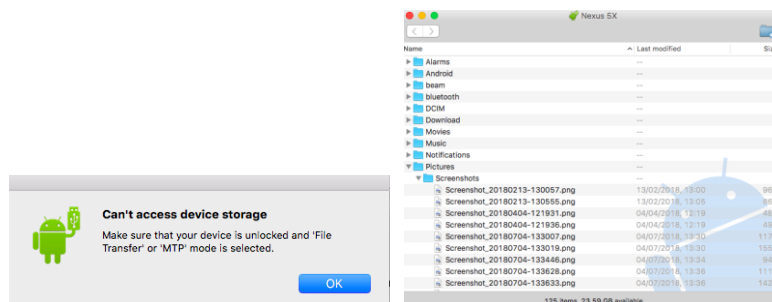
CANCELAR

Sistema Android
Cargando este dispositivo por USB
Toca para ver más opciones.

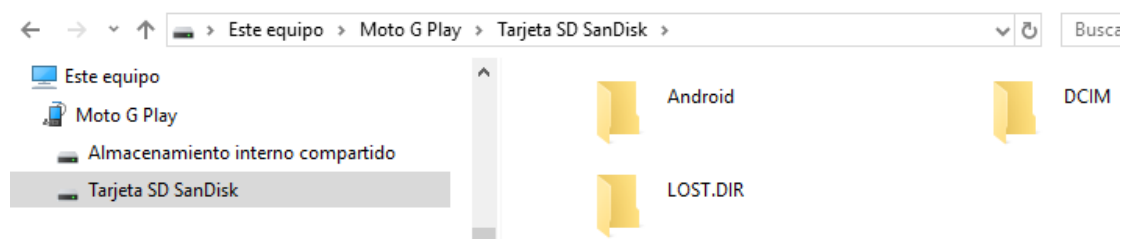
Nota: En Android 7.x ya no está disponible la selección del método de transferencia de datos a través del menú "Ajustes - [Dispositivo] Almacenamiento - [...] - Conexión USB a ordenador" que sí existía en Android 6.x.

1377. Es importante reseñar que, en ocasiones, el uso de un cable USB defectuoso puede provocar que no estén disponibles las opciones de transferencia de datos. Ante un problema de este tipo, se recomienda probar un cable diferente como primera opción.

1378. Para poder acceder a los contenidos del almacenamiento interno desde un ordenador, es necesario desbloquear el dispositivo móvil en al menos en una ocasión desde que se haya conectado el cable USB. A partir de ese momento, es posible disponer de acceso a los contenidos en cualquier momento, incluso aunque el dispositivo móvil se encuentre de nuevo bloqueado⁹¹:



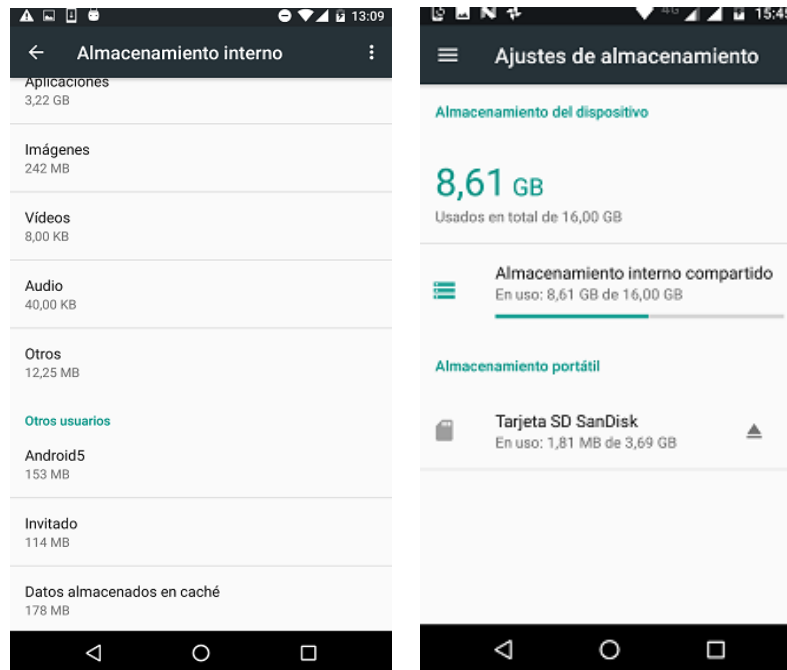
1379. Cuando se selecciona el método "Transferir datos", la conexión entre el ordenador y el dispositivo móvil se llevará a cabo mediante el protocolo MTP, de modo que no se le asigna una unidad independiente (por ejemplo, en Windows no se asignará una letra de unidad ni al almacenamiento interno ni a la tarjeta SD externa), sino que el sistema operativo del ordenador mapeará los contenidos bajo un nuevo dispositivo MTP.



1380. Si se conecta una unidad de almacenamiento USB a través de la conexión USB OTG del dispositivo móvil, Android montará dicha unidad para poder acceder y transferir datos desde y hacia ella.

1381. En el dispositivo móvil Nexus 5X empleado para la elaboración de la presente guía, que no dispone de una ranura (o slot) para una tarjeta de almacenamiento externo (o tarjeta SD) física o real, la unidad de almacenamiento compartida con el ordenador corresponde a una sección denominada "Almacenamiento interno compartido" (imagen inferior izquierda). En un dispositivo que sí la soporte, se verá un apartado para el almacenamiento interno y otro para el externo o tarjeta SD (imagen inferior derecha):

⁹¹ La captura de pantalla que ilustra este comportamiento corresponde a un equipo macOS.



1382. El acceso a los contenidos del almacenamiento (tanto externo como interno) del dispositivo móvil por MTP permite efectuar operaciones de lectura y escritura sobre ellos.
1383. Debe prestarse especial atención a la configuración de seguridad del ordenador, y en concreto, a los mecanismos de auto-ejecución de Windows (autorun y autoplay), para evitar la propagación de software malicioso (como por ejemplo el troyano Ambler u otros códigos dañinos, como en el caso de Vodafone y Android en 2010 [Ref.- 400] desde la tarjeta de almacenamiento externa del dispositivo móvil compartida con el ordenador.
1384. Desde el punto de vista de seguridad, se recomienda mantener la opción de configuración de USB al modo de solo carga a menos que sea necesario realizar una operación que requiera disponer de acceso de datos, y volver al modo carga inmediatamente después.

13.2. CONEXIÓN DE RED COMPARTIDA POR USB (*TETHERING*)

1385. La opción de conexión compartida, o *tethering*, permite compartir la conexión de datos de telefonía móvil (2/3/4G) o Wi-Fi del terminal con un ordenador. Desde el punto de vista de seguridad se recomienda hacer uso de *tethering* mediante cable USB, en lugar de mediante Bluetooth (en particular) o de Wi-Fi [Ref.- 94].
1386. Las capacidades de *tethering* (tanto por USB como por Bluetooth, ver apartado "15.7. Conexión de red compartida por Bluetooth") y de punto de acceso Wi-Fi (ver apartado "16.6. Punto de acceso o zona Wi-Fi") pueden ser utilizadas simultáneamente en Android, proporcionando conexión de red a otros dispositivos u ordenadores a la vez tanto por USB como por Bluetooth o por Wi-Fi:



1387. Android proporciona capacidades nativas de *tethering* para Windows Vista/7/8/10 y Linux, pero no así para macOS⁹². Para hacer uso de estas capacidades es necesario conectar el dispositivo móvil al ordenador mediante USB y acceder al menú "Ajustes - ... Más - [Conexiones Inalámbricas y Redes] - Compartir Internet y zona Wi-Fi", y en concreto a la opción "Compartir por USB".

1388. A través de la opción "Compartir por USB" (ver imagen superior) es posible habilitar el servicio de *tethering*, de forma que el ordenador utilice el dispositivo móvil como su conexión de red.

Nota: Aunque en versiones anteriores de Android el habilitar la conexión de red por USB o *tethering* deshabilitaba el modo de depuración USB de las opciones de desarrollo, en Android 7.x no se observa este comportamiento.

1389. Tras habilitarse la compartición mediante *tethering*, se crea un interfaz de red específico para la conexión en el dispositivo móvil, `rndis0`, cuyos parámetros se pueden obtener a través del comando "adb shell ip addr":

```
rndis0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
link/ether aa:4b:96:36:91:18 brd ff:ff:ff:ff:ff:ff
inet 192.168.42.129/24 brd 192.168.42.255 scope global rndis0
valid_lft forever preferred_lft forever
inet6 fe80::a84b:96ff:fe36:9118/64 scope link
valid_lft forever preferred_lft forever
```

1390. Por su parte, en el equipo que comparte la conexión se creará también el interfaz "Remote NDIS" correspondiente. Por ejemplo, en un ordenador con Windows 10, el interfaz de red creado será de tipo "Remote NDIS based Internet Sharing Service", y la dirección IP dinámica (DHCP) la proporcionará el dispositivo móvil. Se emplea una clase C (/24) (en el siguiente ejemplo, la 192.168.42.235):

```
Adaptador de Ethernet Ethernet 3:
Sufijo DNS específico para la conexión. . . :
Dirección IPv4. . . . . : 192.168.42.235
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.42.129
```

⁹² Existe una solución basada en drivers de terceros para macOS High Sierra, que queda fuera del ámbito de la presente guía.



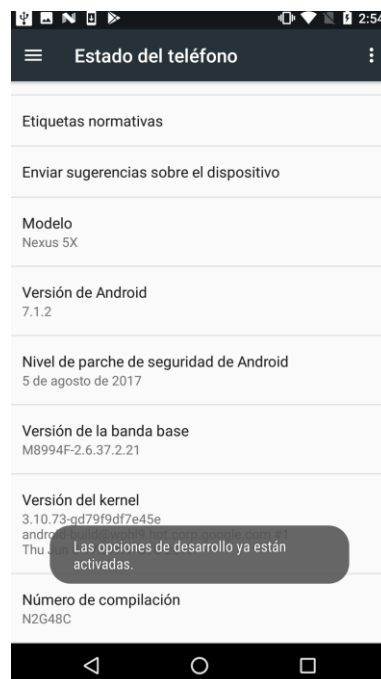
1391. Este interfaz de red NDIS del ordenador tiene asociada una dirección MAC con el OUI "82:77:95" (por ejemplo), que corresponde a una dirección MAC administrada localmente y, por tanto, no asignada a ningún vendedor públicamente.
1392. Se recomienda modificar y restringir la configuración de este interfaz de red para habilitar únicamente los protocolos y servicios que se consideren necesarios en el ordenador.
1393. El dispositivo móvil, con dirección IP 192.168.42.129, actúa de gateway, router o pasarela por defecto, de servidor DHCP, y de servidor DNS, y es alcanzable mediante ping (ICMP).

Nota: El direccionamiento IP empleado por Android para la conexión de red compartida a través de USB (192.168.42.x/24) no puede ser modificado salvo en dispositivos móviles Android *rooteados* [Ref.- 99].

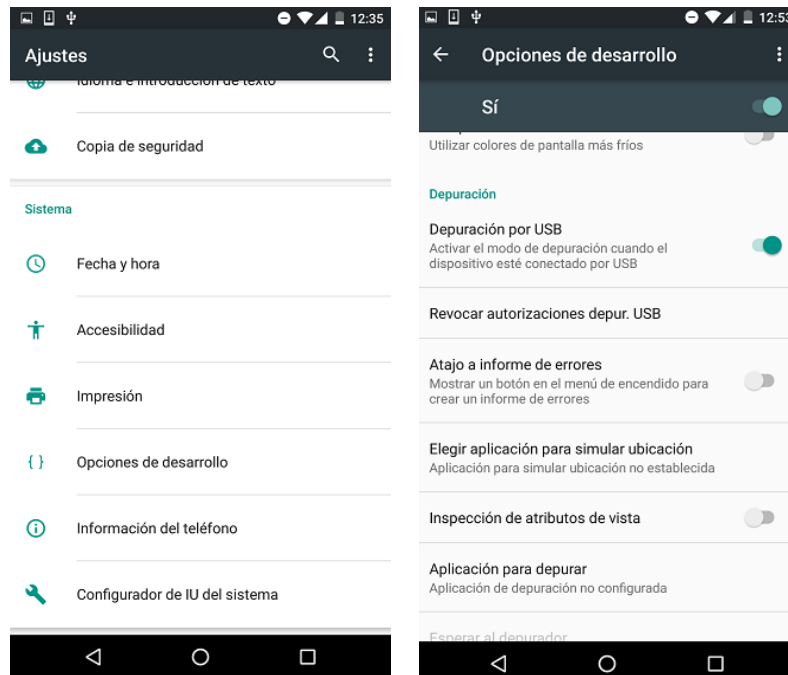
1394. La dirección MAC asociada al interfaz de red empleado para proporcionar la conexión de red compartida a través de USB por el dispositivo móvil tiene asociado un OUI que corresponde a una dirección MAC administrada localmente (al igual que ocurría en el interfaz de red del ordenador al que proporciona servicio) y, por tanto, no asignada a ningún vendedor públicamente.
1395. Un análisis de los puertos TCP/IP y servicios disponibles en el interfaz de red del dispositivo móvil empleado para proporcionar la conexión de red compartida a través de USB permite obtener las siguientes conclusiones:
- El único puerto TCP abierto es el correspondiente al servidor DNS (53/tcp). El servidor DNS es identificado por nmap como "dnsmasq 2.51".
 - Los únicos puertos UDP abiertos son los correspondientes al servidor DNS (53/udp) y al servidor DHCP (67/udp).
1396. Es posible finalizar la compartición de la conexión de datos deshabilitando la opción "Anclaje de USB" habilitada previamente, opción recomendada desde el punto de vista de seguridad una vez no se desea disponer de la conexión de red compartida.

13.3. OPCIONES DE DESARROLLO Y DEPURACIÓN USB

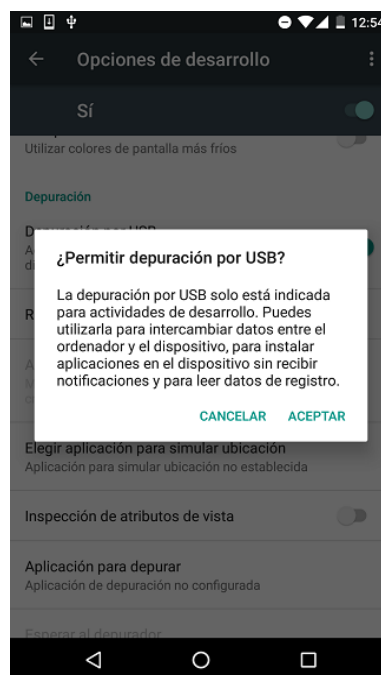
1397. Android proporciona acceso a funciones avanzadas y de bajo nivel a través del puerto USB desde un ordenador, por ejemplo, mediante el *Android Debug Bridge* (ADB) [Ref.- 110]. Estas capacidades son normalmente empleadas por los desarrolladores de apps para Android, con el objetivo de llevar a cabo tareas de depuración, mediante el kit de desarrollo de software de Android, SDK (Software Development Kit).
1398. Desde la versión 4.2 de Android, el menú que proporciona el acceso para habilitar las opciones de desarrollo está oculto por defecto.
1399. Para habilitar dicho menú es necesario acceder al menú "Ajustes [Sistema] - Información del teléfono", y pulsar en siete ocasiones sobre la opción "Número de compilación", disponible en la parte inferior:



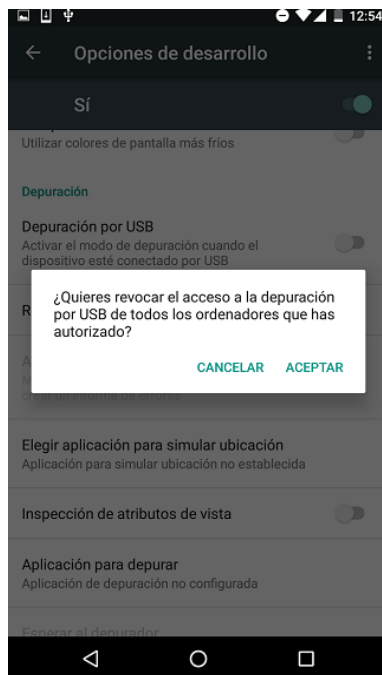
1400. Al completar las pulsaciones, aparecerá un nuevo menú, "Opciones de desarrollo", en la sección de "Ajustes [Sistema]", antes de la opción "Información del teléfono":



1401. A través de este nuevo menú es posible disponer de acceso a múltiples opciones para el desarrollo y depuración de apps en Android, incluyendo la opción "Depuración USB", que permite el acceso a través de USB a las opciones avanzadas de depuración del dispositivo móvil:



1402. Desde el punto de vista de seguridad, se recomienda habilitar las opciones de desarrollo en el dispositivo móvil sólo para aquellas acciones que lo requieran (por ejemplo, realizar un backup a través de "ADB"), y deshabilitarlas una vez realizada la acción que llevó a su activación. Tras deshabilitarlas, se recomienda lanzar la opción "Revocar autorizaciones depur. USB" para que se eliminen del dispositivo las huellas digitales, o asociaciones, correspondientes al ordenador conectado:



1403. Alternativamente, es posible conectarse al dispositivo móvil a través de TCP/IP y poder ejecutar comandos utilizando el protocolo ADB sin necesidad de disponer de una conexión USB física. Es necesario que el dispositivo móvil y el equipo desde el que se va a efectuar la conexión se encuentren en la misma red Wi-Fi, para que ambos puedan comunicarse con el otro.
1404. Para ello, se requiere conectar el dispositivo por USB en una primera ocasión y ejecutar el comando "*adb tcpip <puerto>*" (típicamente, *<puerto>* toma el valor 5555). Una vez se constate que el demonio de ADB se ha reiniciado en modo TCP, se realizará la conexión al dispositivo mediante el comando "*adb connect <IP_del_dispositivo>*" (para obtener la IP del dispositivo, consultar el apartado "19. Comunicaciones TCP/IP").

```
C:\> adb tcpip 5555
restarting in TCP mode port: 5555
C:\> adb connect 172.16.16.123
connected to 172.16.16.123:5555
C:\> adb devices
List of devices attached
172.16.16.123:5555      device
C:\> adb shell
shell@bullhead:/exit
```

1405. Para finalizar la conexión con el dispositivo, se ejecutará el comando "*adb kill-server*". Para volver a las capacidades de depuración por USB, se puede ejecutar "*adb usb*".
1406. Debido a que no es posible restringir las comunicaciones entrantes al dispositivo móvil filtrando por la IP de origen, al habilitar la conexión ADB a través de TCP/IP se está permitiendo que cualquier equipo con acceso a la red Wi-Fi pueda conectarse al dispositivo móvil y enviarle comandos a través de ADB. Esto supone un riesgo importante de cara a la seguridad, por lo que se recomienda no activar el acceso ADB por TCP/IP a menos que exista un motivo justificado. Una vez realizadas las operaciones requeridas, se

ejecutará el comando "*adb kill-server*", junto a "*adb usb*", para evitar que el puerto ADB quede abierto y un tercero pueda conectarse a él.

13.4. RELACIONES DE CONFIANZA A TRAVÉS DE USB

1407. Desde la versión 4.2.2 de Android (Jelly Bean), antes de poder hacer uso de estas capacidades de depuración USB, es necesario establecer una relación de confianza entre el dispositivo móvil y el ordenador conectado a través del puerto USB, permitiendo acceso al terminal a través de una conexión USB de depuración más segura, por ejemplo, mediante ADB [Ref.- 146].

1408. El establecimiento de relaciones de confianza a través de USB en Android se ha implementado mediante claves RSA de 2.048 bits. Estas claves son almacenadas en el ordenador bajo "\$HOME/.android" en Linux y macOS, o "%USERPROFILE%/.android" en Windows, en los ficheros "adbkey" (clave privada, en formato PEM) y "adbkey.pub" (clave pública, en base64).

1409. El dispositivo móvil Android, una vez el usuario acepta una clave de forma permanente, la almacena en el fichero "/data/misc/adb/adb_keys", añadiendo una nueva línea (o entrada) en este fichero correspondiente a la clave contenida en el fichero "adbkey.pub" del ordenador al que se ha conectado por USB:

```
root@movil:/data/misc/adb # ls -l
-rw-r----- system shell 717 2016-05-17 20:00 adb_keys
root@movil:/data/misc/adb # cat adb_keys
QAAAAHOy...<...>...EAAQA= unknown@unknown
```

NOTA: Para acceder a estos ficheros con las claves ADB es necesario disponer de un dispositivo móvil Android *rootado*.

1410. Este mecanismo de protección no puede ser deshabilitado a través de los ajustes de configuración del dispositivo móvil, opción preferida desde el punto de vista de seguridad.

1411. Al habilitar la opción de "Depuración USB", el dispositivo móvil mostrará un mensaje de autorización que deberá ser aprobado por el usuario para poder establecer la conexión a través del puerto USB:

¿Permitir depuración por USB?
La huella digital de tu clave RSA es:
80:A0:EA:15:F6:EF:6C:53:AE:53:79:F9:EC:
93:74:36
☐ Permitir siempre desde este ordenador

CANCELAR ACEPTAR

1412. El mensaje de autorización muestra el hash MD5 de la clave pública del ordenador con el que se ha establecido la conexión vía USB. Para poder comprobar que el valor mostrado por el dispositivo móvil corresponde con el de la clave pública del ordenador al que está conectado, es necesario obtener el fichero "adbkey.pub" del ordenador y extraer su *hash* MD5. Por ejemplo, en Linux puede emplearse el siguiente comando [Ref.- 146]:

```
$ awk '{print $1}' < adbkey.pub | openssl base64 -A -d -a | \
openssl md5 -c | awk '{print $2}'|tr '[:lower:]' '[:upper:]'
```

1413. En el momento de autorizar la relación de confianza, ésta puede permitirse únicamente para la sesión actual (opción por defecto), o de forma permanente, confiando en ese ordenador para futuras conexiones vía USB, mediante la opción "Permitir siempre desde este ordenador" (deshabilitada por defecto).
1414. Este mecanismo de seguridad asegura que sólo ordenadores previamente autorizados por el usuario podrán establecer una conexión mediante el puerto USB cuando el dispositivo móvil está bloqueado y se desconoce el código de acceso.
1415. Mientras la relación de confianza no ha sido aprobada, el dispositivo aparece como "unauthorized" en la respuesta del comando "adb devices" (el identificador del dispositivo móvil corresponde a su número de serie):

```
C:\> adb devices
List of devices attached
02678922bbb3c3b1    unauthorized
C:\> adb devices
List of devices attached
02678922bbb3c3b1    device
C:\> adb shell
shell@bullhead:/ $
```

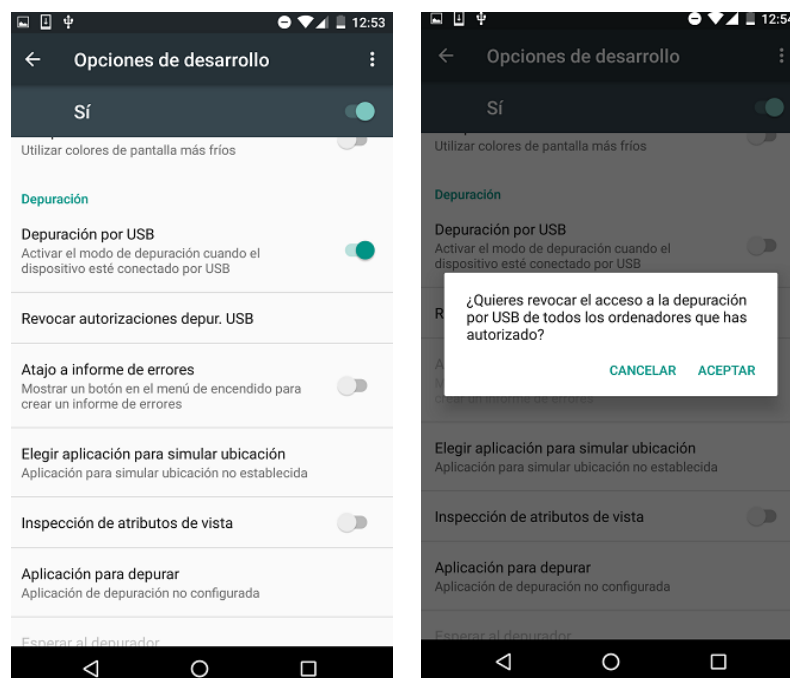
Número de serie
02698922baa3c3b1

Tiempo de actividad
64:55:38

1416. Una vez la relación de confianza ha sido aprobada, el dispositivo aparecerá como disponible ("device") en la respuesta del comando "adb devices" y se podrá acceder a él, mediante ADB.
1417. El dispositivo móvil mostrará una notificación informando de este hecho, y la interacción con la misma abre el menú correspondiente a las "Opciones de desarrollo" para permitir al usuario deshabilitar el modo de depuración:



1418. En el caso de dispositivos móviles Android multi-usuario, únicamente el usuario propietario (o usuario principal) puede autorizar el acceso USB a un ordenador, por lo que para autorizar el mensaje y habilitar la depuración mediante USB es necesario cambiar al usuario principal.
1419. Las relaciones de confianza establecidas previamente por USB pueden ser eliminadas en su conjunto, es decir, no selectivamente, mediante el menú "Ajustes [Sistema] - Opciones de desarrollo [Depuración] - Revocar autorizaciones depur. USB":



1420. Desde el punto de vista de seguridad, se recomienda revocar las autorizaciones otorgadas a un equipo tan pronto se finalice la operación que haya requerido su activación.

14. COMUNICACIONES NFC

14.1. UTILIZACIÓN DE LAS COMUNICACIONES NFC

1421. Android dispone de capacidades de comunicación inalámbricas de corto alcance mediante tecnologías NFC (*Near Field Communication*), en concreto en aquellos dispositivos móviles cuyo *hardware* dispone de un interfaz, módulo o controlador NFC.
1422. NFC permite compartir datos de poco volumen y ancho de banda entre un dispositivo móvil Android y una etiqueta NFC, y también entre dos dispositivos Android. Las etiquetas NFC pueden ser muy simples (permitiendo sólo lectura y/o escritura) o muy complejas (que incluyen entornos de operación para interactuar con ellas).
1423. La mayor parte de las APIs del framework de Android están basadas en el estándar NDEF [Ref.- 278].
1424. El soporte inicial para NFC en Android se introdujo en la versión 2.3, extendiéndose su funcionalidad significativamente en Android 4.4.
1425. Los dispositivos Android soportan tres modos de operación:
- Lectura (*Reader Mode*): permiten la lectura de etiquetas NFC pasivas cuando las apps correspondientes ejecutan en primer plano.
 - P2P (*Point-to-Point*): permiten intercambio básico de datos mediante NFC entre dos dispositivos.

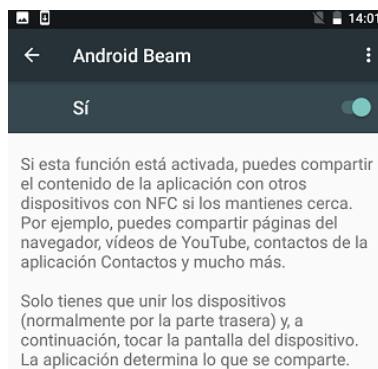
- Emulación de tarjeta: permiten que el dispositivo Android actúe como una tarjeta NFC, que puede accederse desde un lector externo, como un TPV (Terminal Punto de Venta), para la realización de pagos sin contacto.
1426. Por otro lado, las apps pueden declarar el tipo de etiquetas NFC (*tags*) en las que están interesadas, en función de la tecnología de las etiquetas, y cuando el servicio de NFC de Android descubra una nueva etiqueta NFC, enrutará la petición para su procesamiento a la app (o apps) correspondientes previamente registradas [Ref.- 159].
1427. La principal recomendación de seguridad asociada a las comunicaciones NFC en dispositivos móviles es no activar el interfaz inalámbrico NFC salvo en el caso en el que se esté haciendo uso del mismo, evitando así la posibilidad de ataques sobre el *hardware* del interfaz, el driver o la pila de comunicaciones NFC, incluyendo la funcionalidad adicional implementada sobre NFC, como por ejemplo Android Beam [Ref.- 182].
1428. Android no dispone de ninguna indicación en la barra superior de estado, pantalla de inicio o pantalla de bloqueo que indique si el interfaz o controlador NFC está activo, por lo que es necesario acceder a los ajustes de configuración para verificar su estado actual.
1429. El interfaz inalámbrico NFC puede ser desactivado en Android, ya que está habilitado por defecto, a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Más", y en concreto a través de la opción "NFC":



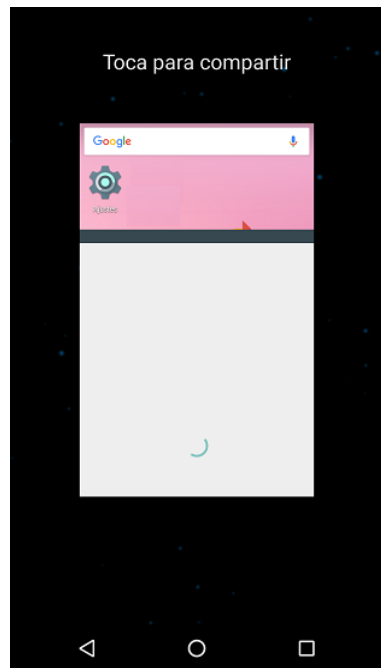
1430. Desafortunadamente, Android no proporciona a través del interfaz de usuario acceso a otros ajustes de configuración más avanzados relacionados con NFC.
1431. Desde el punto de vista de seguridad, y debido al tipo de información sensible y de transacciones y pagos asociada a esta funcionalidad, se recomienda deshabilitar el interfaz NFC y las opciones o servicios adicionales (Android Beam, "Tocar y pagar", etc.), descritas a continuación, salvo que se quiera hacer un uso explícito y controlado de los mismos.

14.2. ANDROID BEAM

1432. Android Beam es una característica que se introdujo en Android 4.1 con objeto de simplificar la compartición de contenidos mediante NFC entre dos dispositivos móviles Android utilizando el modo P2P de NFC.
1433. El término "beam" significa "emitir", pero también "irradiar", lo cual ilustra el concepto de transferencia inalámbrica.
1434. Android Beam proporciona dos modos de compartición:
- "Touch to Beam" (Toca para compartir): permite compartir contenido específico de una app entre dos dispositivos, y depende fundamentalmente de la app que lo comparte. La app receptora en el otro dispositivo actuará en función de dicho contenido.
 - Compartición: para enviar ficheros que no puedan compartirse mediante la función "beam".
1435. Así por ejemplo, la compartición de un enlace por parte del navegador web Chrome provocará que el navegador del dispositivo receptor abra la URL correspondiente automáticamente.
1436. Si el dispositivo receptor no dispone de la app necesaria para procesar el contenido recibido, se abrirá automáticamente la app Google Play para permitir su descarga e instalación.
1437. Android Beam puede ser habilitado a través del interruptor "[SI | NO]" del menú "Ajustes [Conexiones Inalámbricas y Redes] - Más - Android Beam", debiendo estar la opción "NFC" activa previamente:



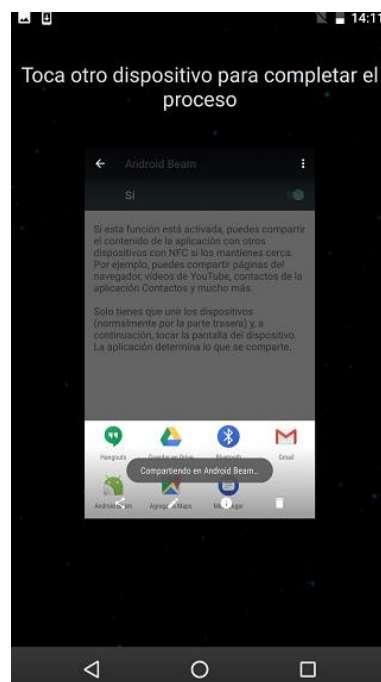
1438. El proceso de transmisión de contenidos se lleva a cabo sin más que aproximar el interfaz NFC de ambos dispositivos móviles, normalmente situado en la parte trasera de los mismos. Al aproximar el dispositivo móvil receptor, una señal acústica confirmará que se ha establecido la conexión NFC, y se solicitará confirmación de la transacción mediante el botón "Toca para compartir":



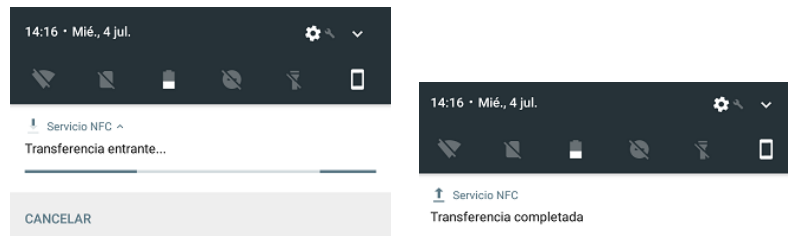
1439. Para ello, y con el objetivo de limitar la realización de transferencias de contenido no intencionadas, es necesario que:

- Los dos dispositivos móviles estén con la pantalla desbloqueada.
- Ambos dispositivos tengan el interfaz NFC activo.
- La funcionalidad Android Beam [Ref.- 182] esté habilitada en ambos dispositivos.

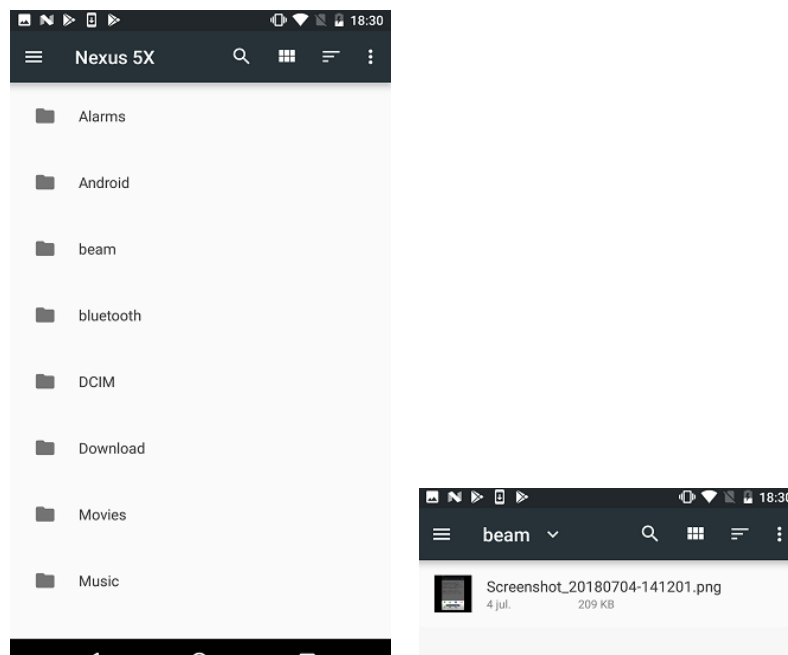
1440. En Android 5.x (Lollipop) se incorporó la funcionalidad de Android Beam al menú "Compartir" (símbolo de grafo de tres puntos), disponible al realizar una pulsación larga sobre un contenido de cualquier aplicación que incorpore el *intent* "ShareActionProvider" [Ref.- 276]:



1441. La flecha en movimiento que se muestra en el menú de notificaciones informa de que la transferencia se ha iniciado. Esta flecha apuntará hacia arriba si el fichero se comparte desde el dispositivo móvil hacia otro dispositivo NFC (*upload*) y apuntará hacia abajo si es un segundo dispositivo móvil el que está compartiendo un elemento con el dispositivo actual (*download*); al desplegarla, se puede ver el progreso de la transacción y cancelarla:



1442. La notificación asociada informa de que el servicio NFC está a cargo de la transferencia.
1443. Cuando se transfieren imágenes por primera vez a través de Android Beam, se creará en el almacenamiento interno del dispositivo móvil receptor una carpeta denominada "beam":



1444. Para que una app pueda realizar una transferencia por NFC, debe solicitar en su fichero *manifest* el permiso NFC ("android.permission.NFC").
1445. Si bien la compartición a través de la funcionalidad Beam requiere que los dos dispositivos implicados estén muy próximos (a menos de 4 ó 5 centímetros aproximadamente), desde el punto de vista de seguridad se recomienda no habilitarla para evitar que contenidos sensibles puedan ser transferidos desde el dispositivo móvil del usuario a otro dispositivo.
1446. Por ejemplo, sería posible una transferencia accidental (o intencionada) de contenidos entre usuarios que viajen en un medio de transporte como

metro o autobús, cuyos dispositivos se encuentren en un bolsillo y se acerquen suficientemente.

14.3. PAGOS (Y OTROS SERVICIOS) MEDIANTE NFC

1447. Desde su versión 4.4 (KitKat), Android dispone de soporte para *Host Card Emulation* (HCE) [Ref.- 181]. Mediante HCE cualquier app en Android puede emular una tarjeta NFC (*smartcard*)⁹³ y por tanto, iniciar transacciones NFC, no siendo necesario obligatoriamente hacer uso del *Secure Element* (SE) y de las *applets* instaladas en el propio SE.
1448. El *Secure Element* (SE) es un componente *hardware* (implementado como el chip de las tarjetas inteligentes, o *smartcards*) que proporciona tanto capacidades de almacenamiento seguro (por ejemplo, de claves de autenticación u otro material sensible), como un entorno de ejecución seguro para la ejecución de código y tareas críticas.
1449. El SE suele estar integrado en los dispositivos móviles actuales como una *Universal Integrated Circuit Card* (UICC, es decir, una tarjeta SIM), embebido en el propio hardware del terminal, o a través de una tarjeta SD especial. Si el dispositivo móvil dispone de soporte para NFC, el SE normalmente está embebido en el terminal y conectado al controlador NFC.
1450. Dentro de la gama de dispositivos móviles Nexus de Google (y posteriormente Pixels), el primer modelo con capacidades NFC fue el Nexus S, seguido de sus sucesores el Galaxy Nexus y el Nexus 4. Todos estos modelos, además de disponer de un interfaz o controlador NFC, disponían de un SE embebido (denominado eSE, *embedded SE*).
1451. Cuando la emulación de tarjeta NFC se realiza a través del SE, la tarjeta a emular se provisiona dentro del SE a través de una app. El controlador NFC del dispositivo dirigirá todos los datos desde el lector NFC hacia el SE.
1452. En versiones anteriores a la 4.4 de Android, sólo era posible hacer uso del modo de emulación de tarjeta si se hacía uso del SE conectado al controlador NFC. Android 4.4 introdujo un método adicional de emulación de tarjetas que no requiere de un SE, y que se denominó "*Host-based card emulation*" (HCE).
1453. Con HCE, las apps actúan como tarjetas sin contacto virtuales (*virtual contactless smartcards*), de forma que los datos se enrutan hacia la CPU en la que ejecuta la app, en lugar de enviar las tramas NFC a un SE.
1454. Debido al soporte para HCE introducido por Android desde la versión 4.4, los nuevos modelos de dispositivos móviles Nexus de Google, como el Nexus 5 o el Nexus 7 (2013), así como los modelos Pixel, no disponen de un SE embebido (en favor de HCE).
1455. Las apps de pago, como por ejemplo apps bancarias, disponen de la posibilidad de registrar su identificador de app NFC (AID) dinámicamente y

⁹³ Android HCE emula tarjetas ISO/IEC 7816 que hacen uso del protocolo ISO/IEC 14443-4 (ISO-DEP) para las transmisiones sin contacto (*contactless*).

de fijar como preferencia el servicio de emulación de tarjeta que se desea utilizar cuando las actividades están ejecutando en segundo plano.

1456. En septiembre de 2011 Google introdujo Google Wallet⁹⁴ en Estados Unidos para la realización de pagos. Esta app no tuvo gran aceptación, principalmente porque era necesario disponer de una adecuada integración entre Android (software) y el controlador NFC del dispositivo móvil (hardware), y de acceso al *Secure Element* (SE) existente en el dispositivo móvil. En muchos modelos de terminal este acceso estaba limitado o restringido, especialmente para la instalación de nuevas *applets* en el SE.

1457. Con la introducción de HCE, Android proporciona una nueva arquitectura más abierta y flexible para realizar pagos NFC, que funciona potencialmente con cualquier dispositivo móvil y que permite a las apps gestionar la información sobre pagos en "la nube" o en cualquier dispositivo móvil.

1458. Estas capacidades permiten a los dispositivos móviles Nexus o Pixel de Google realizar pagos a través del servicio "Tocar y pagar" (Tap & Pay) en tiendas que admiten pagos sin contacto (*contactless*):

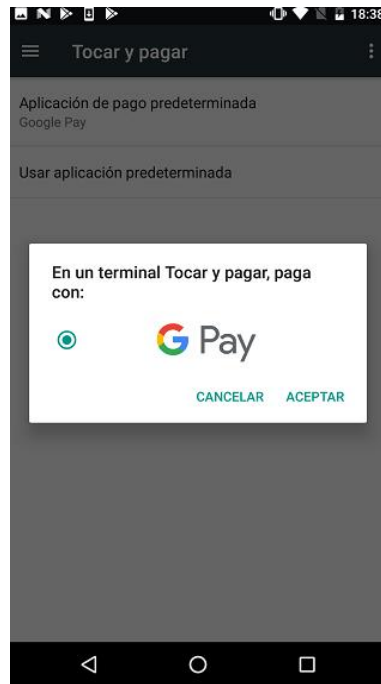


1459. Adicionalmente, NFC puede ser empleado para gestionar las tarjetas de fidelización, tarjetas de transporte, tarjetas de identificación y otros servicios similares.

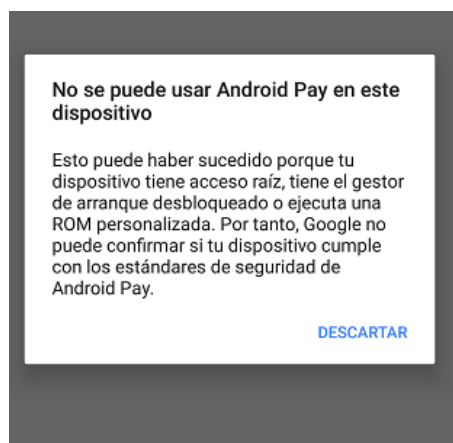
1460. En la versión 6.x de Android, se introdujo el servicio "Android Pay", basado en HCE, que amplió la funcionalidad de Google Wallet. Con Android 7.x, la app "Google Pay", sustituta de Android Pay, es la app predeterminada para la función "Tocar y pagar", y viene instalada por defecto en los dispositivos Nexus.

1461. La funcionalidad de "Tocar y pagar" (Tap & Pay) [Ref.- 183] accesible a través del menú de "Ajustes [Dispositivo] - Tocar y pagar", está activa en Android 7.x por defecto.

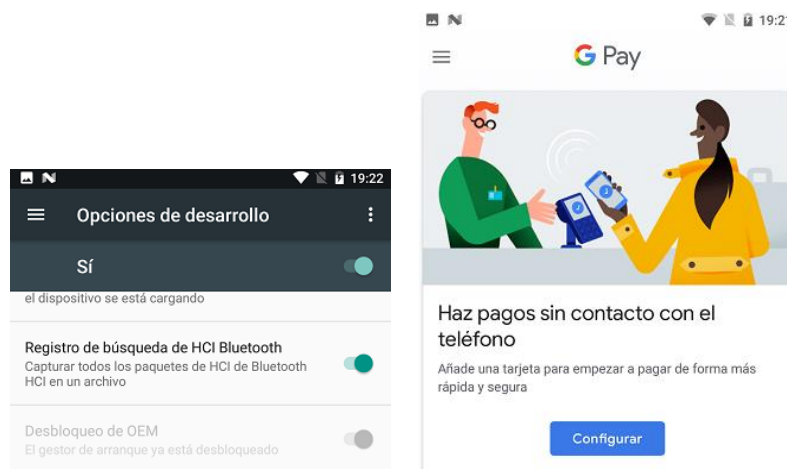
⁹⁴ <https://wallet.google.com>



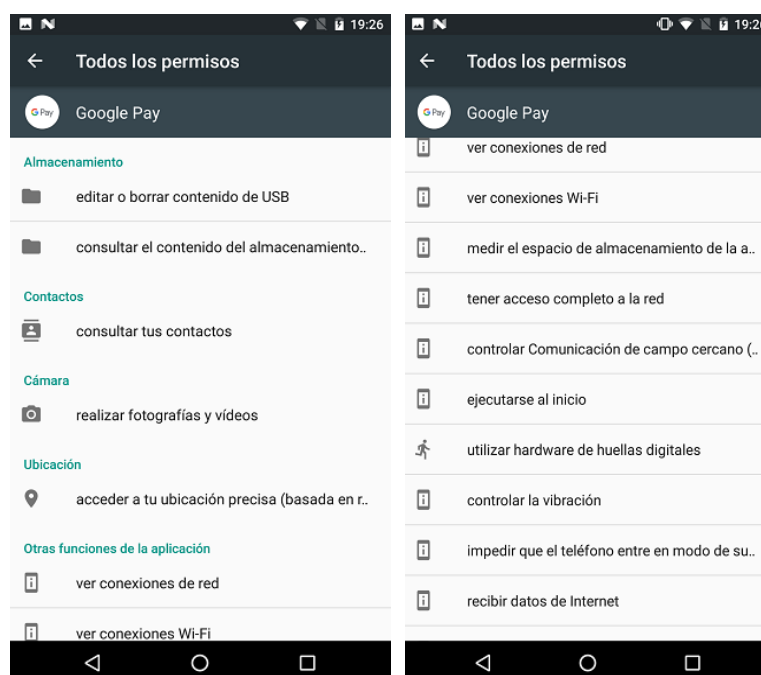
1462. No existe ningún interruptor que controle la activación/desactivación de esta funcionalidad. Únicamente es posible hacer uso del botón "Inhabilitar" asociado a esta app a través del gestor de aplicaciones.
1463. Adicionalmente, el servicio "Tocar y pagar" quedará deshabilitado si se inhabilita el interfaz NFC en Android.
1464. Android Pay no podía ser utilizado en dispositivos móviles Android *rooteados* (indicado como "acceso raíz" en el siguiente mensaje de error), ni tampoco en aquellos que tengan el gestor de arranque desbloqueado, o hagan uso de una versión de firmware (o ROM) de Android personalizada (como, por ejemplo, LineageOS). Estas comprobaciones son llevadas a cabo a través de las librerías del servicio SafetyNet de Google, y Google lleva a cabo cambios en estos requisitos de manera dinámica (por lo que a lo largo del tiempo se ha permitido y se ha denegado el uso de Android Pay, por ejemplo, a dispositivos con el gestor de arranque desbloqueado). Cuando se intentaba configurar Android Pay en un dispositivo móvil que incumplía alguna de estas condiciones, se presentaba el siguiente mensaje de error:



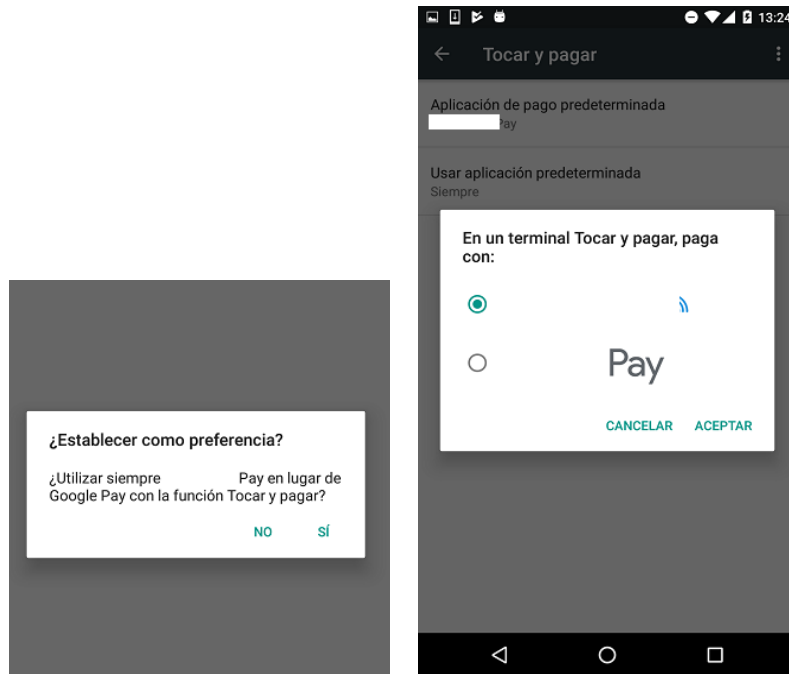
1465. Sin embargo, Google Pay (en el momento de elaboración de la presente guía y con los requisitos actuales de SafetyNet) no impone la restricción para dispositivos que tengan el gestor de arranque desbloqueado, como ilustran las siguientes imágenes:



1466. Adicionalmente, para hacer uso de Google Pay se requiere haber definido un código de acceso para el dispositivo móvil.
1467. Google Pay se incorpora como una app administrador de dispositivo, con capacidades para bloquear el terminal remotamente y para borrar los datos de pago de forma que, si el usuario pierde el dispositivo móvil, no se pueda utilizar el terminal para realizar pagos.
1468. Con el nuevo modelo de permisos de Android introducido en la versión 6.x (consultar apartado "7.1. Modelo de permisos de Android"), el permiso de acceso al interfaz NFC se ve englobado en el apartado "Permisos adicionales", se considera normal, y no se solicita confirmación al usuario para utilizarlo en tiempo de ejecución. Para verificar qué permisos posee una app para el pago vía NFC, se debe acceder a "Ajustes - [Dispositivo] Aplicaciones - Google Pay - Permisos [...] - Todos los permisos:



1469. El acceso al interfaz NFC se refleja como "controlar Comunicación de campo cercano (NFC)".
1470. Si se instala una nueva app para la función "Tocar y pagar", la misma podrá solicitar convertirse en la app de pagos por defecto. El usuario podrá elegir en cualquier momento cuál emplear en el apartado "Ajustes - [Dispositivo] Tocar y pagar - Aplicación de pago predeterminada":

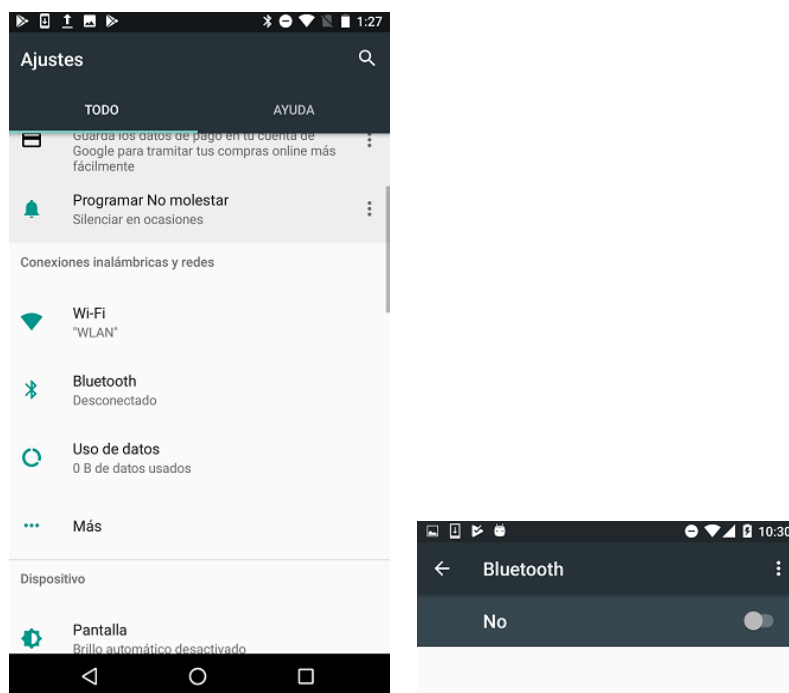


1471. El servicio "Google Pay" pretende unificar los distintos medios de pago (Android Pay y Google Wallet) como parte del servicio ofrecido al usuario a través de su cuenta de Google. El objetivo del servicio es que la configuración de pago esté disponible en cualquier dispositivo en el que se configure la cuenta de usuario de Google, y que se pueda usar tanto para compras vía web a través de Chrome, en apps de Android con compras integradas, en pagos vía NFC en comercios físicos, etc. [Ref.- 255].
1472. A pesar de que la transacción en las operaciones de pago a través de Google Pay y Android Pay no revela la numeración real tarjeta al receptor del pago, sino un número de cuenta virtual que se asocia durante la operación de alta de la tarjeta en el servicio, desde el punto de vista de seguridad no se recomienda hacer uso de estos sistemas sin evaluar las implicaciones de seguridad en la utilización de los mismos.
1473. Aunque más incómodo para el usuario, se recomienda habilitar temporalmente estas capacidades únicamente en el momento de ir a hacer uso de las mismas, volviendo a deshabilitar el interfaz NFC tan pronto se haya completado la transacción o el intercambio de datos mediante esta tecnología inalámbrica.
1474. Adicionalmente, si se desea hacer uso del servicio Google Pay, se recomienda ser extremadamente cauteloso en las medidas de protección de la cuenta de usuario de Google.

15. COMUNICACIONES BLUETOOTH

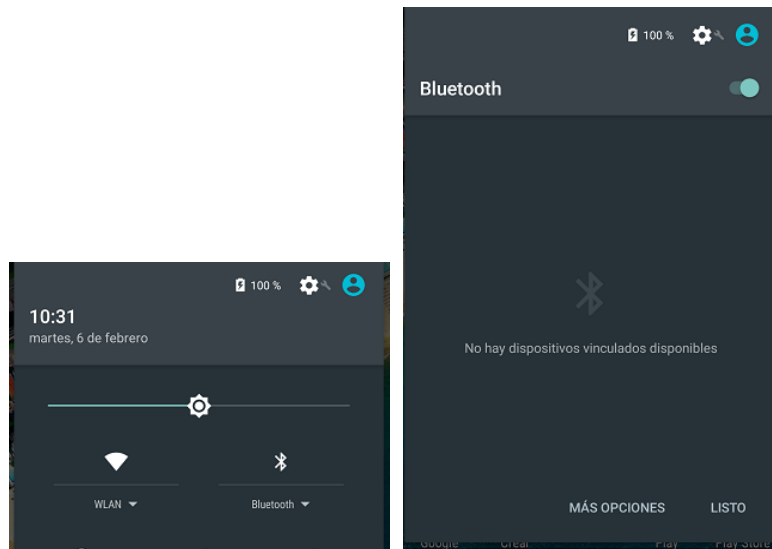
15.1. UTILIZACIÓN DE LAS COMUNICACIONES BLUETOOTH

1475. Desde el punto de vista de seguridad, se recomienda no activar el interfaz inalámbrico Bluetooth en el dispositivo móvil salvo en el caso en el que se esté haciendo uso del mismo, evitando así la posibilidad de ataques sobre el hardware del interfaz, el driver o la pila de comunicaciones Bluetooth, incluyendo los perfiles Bluetooth disponibles [Ref.- 50].

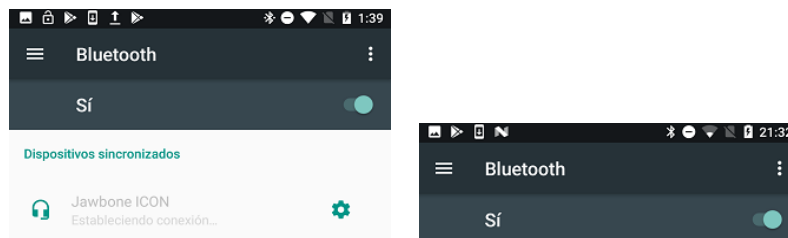


1476. El interfaz inalámbrico Bluetooth puede ser desactivado en Android, estado por defecto, a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Bluetooth"⁹⁵, mediante el botón "SÍ|NO" de la opción "Bluetooth". También es posible activarlo/desactivarlo desde el menú de ajustes rápidos del dispositivo:

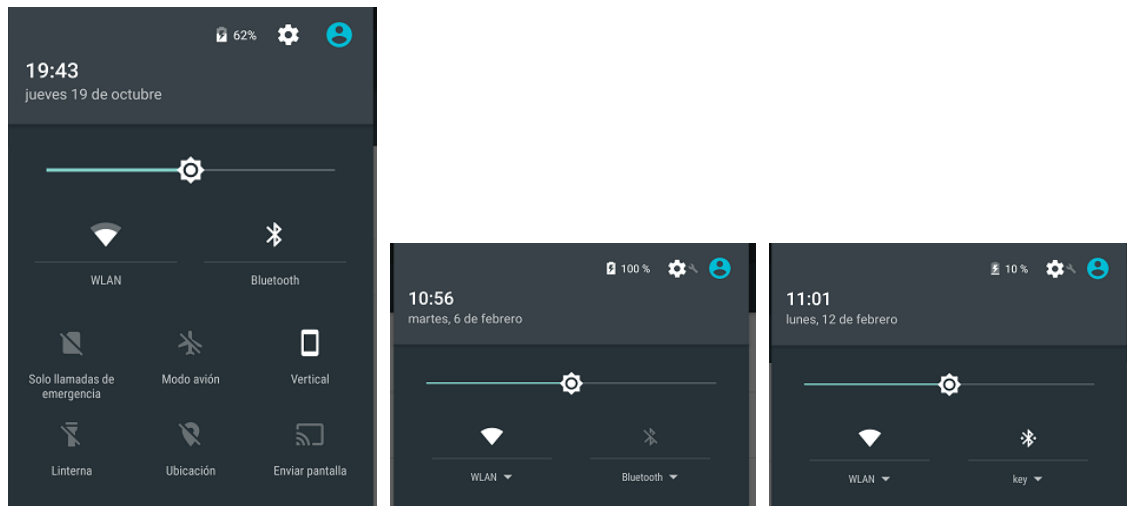
⁹⁵ A partir de este momento, por simplificar, el menú "Ajustes [Conexiones Inalámbricas y Redes] - Bluetooth" se abreviará como "Ajustes - Bluetooth".



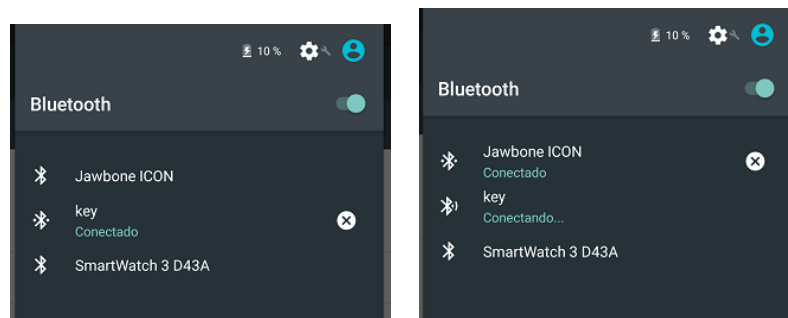
1477. El estado del interfaz Bluetooth puede ser verificado de forma rápida y sencilla por el usuario del dispositivo móvil a través de la barra superior de estado, ya que tan pronto el interfaz está activo, se mostrará el icono de Bluetooth.
1478. Android proporciona información en la barra superior de estado sobre el estado del interfaz Bluetooth, y si existe una conexión establecida actualmente, a través de un conjunto específico de iconos que muestran el logotipo de Bluetooth. Por ejemplo, el icono de Bluetooth tendrá un par de pequeños puntos (uno a cada lado) indicando la existencia de una conexión (ver la barra de notificaciones de la imagen inferior):



1479. Cuando el interfaz Bluetooth está activo pero no hay conexiones establecidas, se mostrará el símbolo de Bluetooth sin los puntos a ambos lados.
1480. Desde el menú de ajustes rápidos también es posible ver el estado actual del interfaz Bluetooth: el interfaz activo pero sin conexiones se muestra en color blanco y el texto "Bluetooth" (imagen inferior izquierda), el interfaz desactivado en color gris y con una línea diagonal (imagen central) y la existencia de alguna conexión con el símbolo de Bluetooth en blanco y un punto a cada uno de sus lados seguido del nombre del dispositivo con el que se ha establecido la conexión (imagen inferior derecha):



1481. Desde el menú de ajustes rápidos es posible gestionar tanto el estado del interfaz Bluetooth (activar/desactivar) como las conexiones con los dispositivos vinculados. El menú muestra el nombre del dispositivo con el que está establecida la conexión (si hay más de una, el nombre que tenga precedencia por orden alfabético). Para ver todas las conexiones activas, se pulsará sobre la flecha situada a la derecha del nombre del dispositivo conectado, lo cual desplegará un menú con todos los dispositivos vinculados:



1482. Cuando se reactiva el interfaz Bluetooth después de haber sido inhabilitado, por defecto Android intentará establecer la conexión con el primero de los dispositivos de su lista de emparejamiento (por orden alfabético) que esté disponible. Si se desea establecer conexiones adicionales, habrá que pulsar manualmente sobre el nombre de los demás dispositivos.
1483. Android mantiene el estado del interfaz Bluetooth tras reiniciar el dispositivo móvil, es decir, si el interfaz Bluetooth estaba activo al apagar el terminal, al encender el dispositivo móvil seguirá activo.
1484. La activación inicial del interfaz Bluetooth se llevará a cabo tan pronto como el dispositivo móvil se haya reiniciado, incluso aunque aún no haya sido desbloqueado por primera vez tras encenderlo (igual que el interfaz Wi-Fi).
1485. Igualmente, al salir del modo avión el estado del interfaz Bluetooth será restaurado, activándose de nuevo en el caso de haber estado activo previamente (antes de activar el modo avión; igual que el interfaz Wi-Fi).

1486. El interfaz Bluetooth también permanece activo cuando el dispositivo móvil está en espera o *stand-by*, es decir, encendido pero con la pantalla apagada y bloqueado.

Nota: Tras un restablecimiento de fábrica, o *hard-reset*, el interfaz Bluetooth del dispositivo móvil empleado para la elaboración de la presente guía está deshabilitado por defecto. Una vez el interfaz es habilitado por el usuario, éste responde a peticiones Bluetooth (en modo oculto (o no visible) por defecto).

1487. La dirección del interfaz Bluetooth, BD_ADDR (Bluetooth Device Address) está disponible a través del menú "Ajustes - [Sistema] Información del teléfono - Estado", y en concreto, en el campo "Dirección de Bluetooth". La dirección no es visible, se muestra el mensaje "No disponible" en su lugar (a diferencia del interfaz Wi-Fi), si el interfaz Bluetooth no se encuentra activo:

Dirección IP fe80: 172.1	Dirección MAC de Wi-Fi 2c:59:
Dirección MAC de Wi-Fi 2c:59:	Dirección de Bluetooth 80:5a:04:
Dirección de Bluetooth No disponible	Número de serie 0265

Nota: La parte de la dirección Bluetooth (BD_ADDR) asociada al fabricante, NAP+UAP (primeros tres *bytes* de la dirección) puede permitir identificar el tipo de dispositivo móvil. En el ejemplo analizado, el valor "64:89:9A" está asignado por el IEEE (<http://standards.ieee.org/regauth/oui/index.shtml>) a "LG":

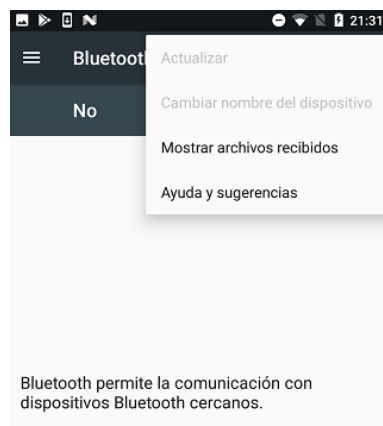
80-5A-04	(hex)	LG Electronics (Mobile Communications)
805A04	(base 16)	60-39, Gasan-dong, Geumcheon-gu
		Seoul Seoul 153-801
		KOREA, REPUBLIC OF

1488. Con objeto de proteger la privacidad del usuario, en Android 6.x se bloqueó el acceso directo a los identificadores hardware del dispositivo móvil de cara a las apps que no son de sistema; por ejemplo, para las direcciones MAC asociadas a las APIs de Wi-Fi y Bluetooth [Ref.- 602], si una app solicita conocer una de estas direcciones, en lugar de la dirección MAC real, Android devolverá el valor "02:00:00:00:00:00".

1489. Complementariamente, y también con el objetivo de proteger la privacidad del usuario, cuando un dispositivo móvil inicia un escaneo Bluetooth de fondo o *background* (este aspecto también aplica a los escaneos Wi-Fi), la operación se lleva a cabo empleando una dirección MAC origen aleatoria, por lo que los dispositivos externos no pueden identificar de manera directa la dirección MAC real del dispositivo móvil. Sin embargo, debe tenerse en cuenta que, si posteriormente, el dispositivo móvil establece una conexión con un dispositivo externo, el dispositivo hará uso de su dirección MAC real, desvelándose su valor [Ref.- 602].

15.2. CONFIGURACIÓN GENERAL DE BLUETOOTH

1490. Debe tenerse en cuenta que, al igual que para el interfaz Wi-Fi, para poder llevar a cabo la configuración de la pila de comunicaciones Bluetooth a través del interfaz de usuario del dispositivo móvil es obligatorio activar el interfaz (como se muestra en la siguiente imagen, salvo para ver los archivos recibidos) por lo que se recomienda activar el interfaz y realizar las modificaciones de la configuración en un entorno seguro, para evitar potenciales ataques cuando los mecanismos de seguridad no han sido aún aplicados:



1491. Desde la pantalla de activación del interfaz Bluetooth se accede a las opciones avanzadas a través del botón con tres puntos "...". Estas opciones permiten configurar el nombre del dispositivo móvil empleado para las conexiones Bluetooth, iniciar la búsqueda de dispositivos en rango y mostrar los archivos recibidos por Bluetooth.



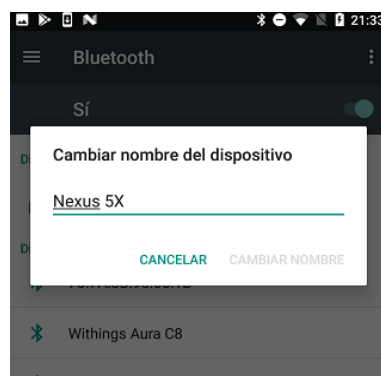
1492. La posibilidad de configurar el estado del interfaz Bluetooth (oculto o visible) y el tiempo de visibilidad, o de buscar otros dispositivos Bluetooth con los que conectarse, ya no se puede gestionar desde las opciones avanzadas, al contrario de lo que ocurría en versiones previas de Android.

1493. Desde Android 5.x, el dispositivo móvil está visible desde el punto de vista de Bluetooth mientras permanezca en la pantalla de configuración de Bluetooth, tal como indica el mensaje disponible en la parte inferior de la

misma. Igualmente, el dispositivo móvil automáticamente buscará otros dispositivos Bluetooth una vez se encuentre en esta pantalla.

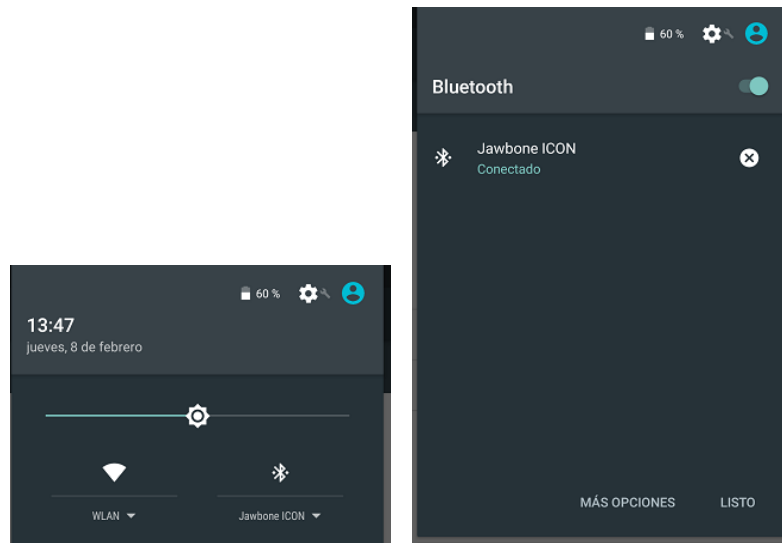
Nota: En ocasiones puede ser necesario deshabilitar y volver a habilitar el interfaz Bluetooth para que el emparejamiento entre dispositivos sea exitoso. Si se detectan problemas en el proceso de emparejamiento, además de consultar las instrucciones de emparejamiento del otro dispositivo, conviene probar a deshabilitar y habilitar el interfaz Bluetooth del dispositivo móvil.

1494. En caso de no especificar ningún nombre para Bluetooth, el valor por defecto empleado como nombre de dispositivo móvil para las conexiones Bluetooth por Android es "Nexus 5X" (correspondiente al tipo y modelo exacto de dispositivo móvil empleado para la elaboración de la presente guía), tal como se indica en el menú correspondiente a "Cambiar nombre del dispositivo" en los ajustes de Bluetooth:



1495. Debido a que el nombre por defecto desvela el tipo de dispositivo móvil empleado, se recomienda modificarlo por un valor que no revele detalles ni del dispositivo móvil (fabricante o modelo) ni del propietario (evitando tanto referencias al nombre de la persona como de la organización asociadas al mismo), como por ejemplo "bt0001" ("Bluetooth device 0001").
1496. En el caso de activar el interfaz Bluetooth para hacer uso de esta tecnología, se recomienda configurar el mismo en modo oculto o no visible, con el objetivo de no desvelar su presencia frente a las operaciones de búsqueda de otros dispositivos Bluetooth.
1497. Por defecto, al activar el interfaz Bluetooth, éste se encuentra en modo oculto o no visible, opción recomendada desde el punto de vista de seguridad. Sin embargo, desde Android 5.x, al permanecer en la pantalla de configuración de Bluetooth, el dispositivo móvil estará visible.
1498. Para activar el interfaz Bluetooth en modo oculto, se recomienda no entrar en el menú de ajustes de Bluetooth, sino utilizar el menú de ajustes rápidos del dispositivo móvil.
1499. La cantidad de tiempo que el dispositivo permanecerá en estado visible depende del tiempo que el usuario permanezca en la pantalla de configuración de Bluetooth.
1500. Se recomienda minimizar la visibilidad del dispositivo al mínimo tiempo necesario para realizar el emparejamiento, en caso de establecer la conexión

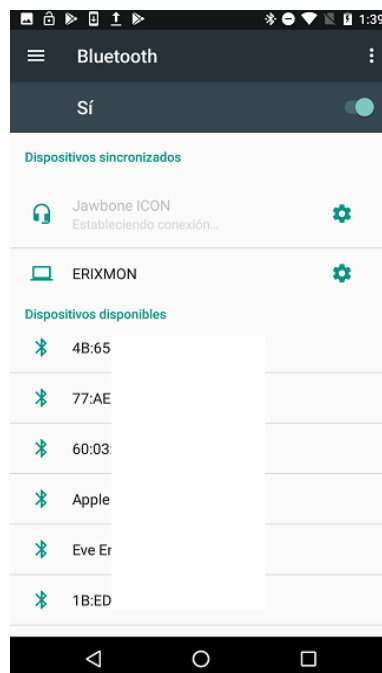
desde el propio dispositivo móvil (y no desde el otro dispositivo Bluetooth, ya que sólo es necesario que uno de los dos dispositivos se encuentre en estado visible para establecer el emparejamiento inicial).



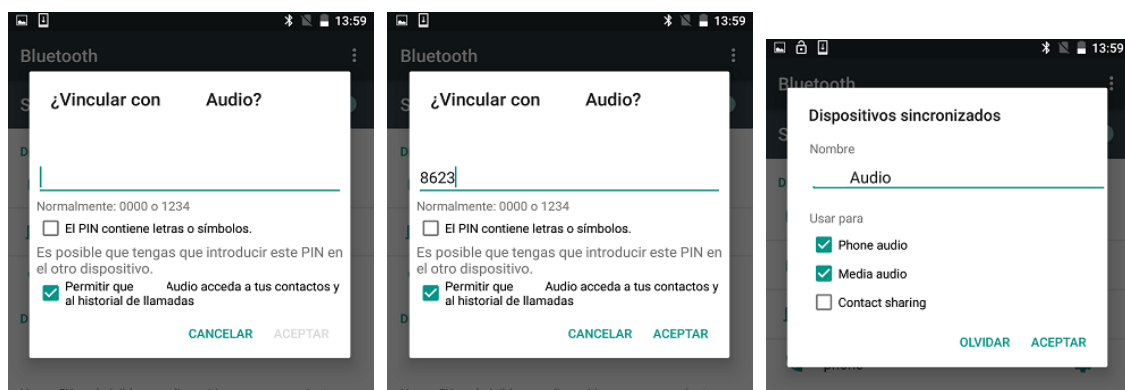
1501. Al llevar a cabo conexiones Bluetooth y otras operaciones de emparejamiento con otros dispositivos desde el dispositivo móvil, al estar en la pantalla de configuración, se obliga a que el interfaz Bluetooth del dispositivo móvil permanezca en modo visible.

15.3. DISPOSITIVOS BLUETOOTH EMPAREJADOS

1502. Android lleva a cabo de forma automática búsquedas de dispositivos Bluetooth visibles (por ejemplo, al habilitar el interfaz Bluetooth), proceso indicado por una rueda giratoria en la parte derecha junto al texto "Dispositivos disponibles", mostrando los mismos bajo dicha sección de la pantalla de configuración de Bluetooth:

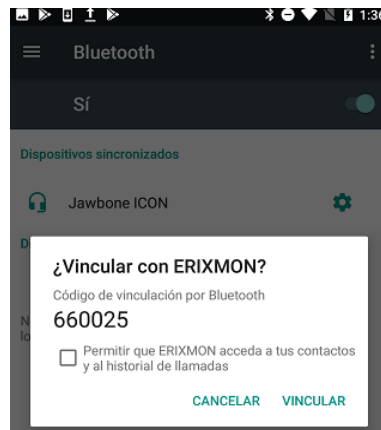


1503. La operación de búsqueda puede llevarse a cabo también de forma manual, o, alternativamente, saliendo de la sección de ajustes de Bluetooth y volviendo a entrar en la misma.
1504. Para realizar una búsqueda o descubrimiento de otros dispositivos Bluetooth cercanos desde la pantalla de configuración, es necesario emplear las opciones avanzadas disponibles desde el botón con tres puntos "...", y en concreto, la opción "Actualizar".
1505. Mediante la selección de un dispositivo visible de la lista es posible realizar el emparejamiento con el mismo. Android establecerá la comunicación Bluetooth con el otro dispositivo y llevará a cabo el proceso de emparejamiento, solicitando al usuario el PIN o contraseña Bluetooth si el otro dispositivo así lo requiere.
1506. En función de la versión de Bluetooth que soporte el dispositivo con el que se va a establecer la vinculación, la operación puede requerir o no la introducción de un código de emparejamiento (PIN o contraseña).
1507. Hasta la versión o especificación 2.0 de Bluetooth (inclusive), el emparejamiento se llevaba a cabo con un código de emparejamiento compartido (*Legacy Pairing*). Normalmente, es el fabricante el que define el código de emparejamiento, que puede ser fijo, como en el caso de dispositivos que no disponen de pantalla o interfaz de usuario, o variar como, por ejemplo, en dispositivos que tienen la capacidad de mostrar un código por pantalla (por ejemplo, una radio de un vehículo). En este caso, el usuario deberá de introducir en Android el código mostrado por el otro dispositivo con el que se vinculará.
1508. En el emparejamiento "legacy", el usuario deberá introducir el PIN o contraseña Bluetooth en el dispositivo móvil. Dicho código de emparejamiento se mostrará en la pantalla al ser introducida por el usuario, sin disponer Android de ninguna opción para ocultar su valor, por lo que debe prestarse especial atención a que nadie, aparte del usuario, disponga de acceso visual a la pantalla durante el proceso de emparejamiento.



1509. A partir de la especificación 2.1 de Bluetooth (core version 2.1) [Ref.-256], se introdujo en los dispositivos un nuevo método de emparejamiento mediante SSP (*Secure Simple Pairing*), que introduce protección frente a ataques de interceptación (*eavesdropping*) y MitM (*Man-in-the-Middle*). SSP define cuatro métodos o modos de asociación:

- *Numeric comparison* (comparación numérica): este método se emplea cuando ambos dispositivos tienen una pantalla en la que mostrar un código de 6 dígitos y un mecanismo para aceptar/rechazar el código visualizado en dichas pantallas. Si el código coincide en ambos extremos de la comunicación, se seleccionará la opción correspondiente para aceptar el emparejamiento. Este escenario es típico en dispositivos móviles (como el empleado en la presente guía), *smartwatches*, ordenadores, etc.



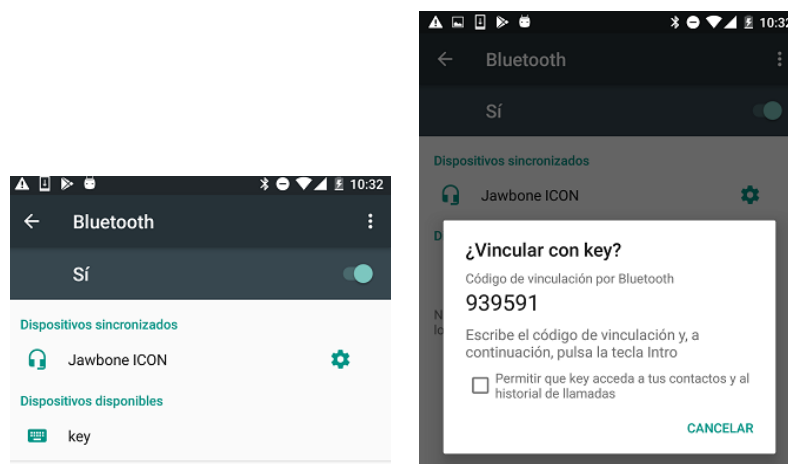
- "Just Works": este método se emplea cuando al menos uno de los dos dispositivos carece de pantalla o de mecanismo de aceptación de un código. Internamente, los dos dispositivos intercambiarán un número, de forma similar al caso anterior, pero el usuario no verá dicho número y no se le solicitará confirmación para el emparejamiento. Este escenario es típico en el emparejamiento entre un dispositivo móvil y un auricular inalámbrico o manos libres. Este método no ofrece protección frente a ataques MitM, aunque sí frente a ataques de interceptación.



- *Out-of-Band* (OOB, o mecanismo de comunicación fuera de banda): el propósito de este método es que el proceso de descubrimiento de dispositivos se realice a través de un canal alternativo, que también se

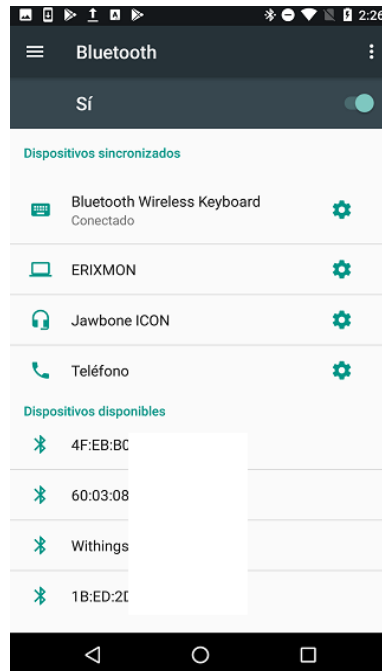
emplearía para el intercambio de los códigos de emparejamiento. Este canal debería ser inmune a ataques MitM. Por ejemplo, a través de NFC (o a través de Wi-Fi), enfrentando físicamente los dos dispositivos se podría realizar el intercambio de la información de descubrimiento (por ejemplo, las direcciones Bluetooth) y del material criptográfico. Uno de los dispositivos establecería la conexión y se emplearía la información obtenida previamente para formalizar el emparejamiento con el otro dispositivo.

- *Passkey entry* (introducción de clave): en este método, típico de escenarios en los que uno de los dispositivos sí tiene capacidades para introducir datos pero no para mostrarlos (por ejemplo, un teclado Bluetooth que se conecta a un ordenador), el dispositivo con pantalla mostrará al usuario un código de 6 dígitos y solicitará al usuario su introducción en el otro dispositivo, permitiendo la vinculación si ambos valores coinciden. La principal diferencia con el método de "legacy pairing" de la especificación 2.0 y anteriores de Bluetooth es que el código de emparejamiento es independiente del algoritmo de seguridad, por lo que conocer dicho código no permite descifrar los datos intercambiados entre ambos dispositivos.

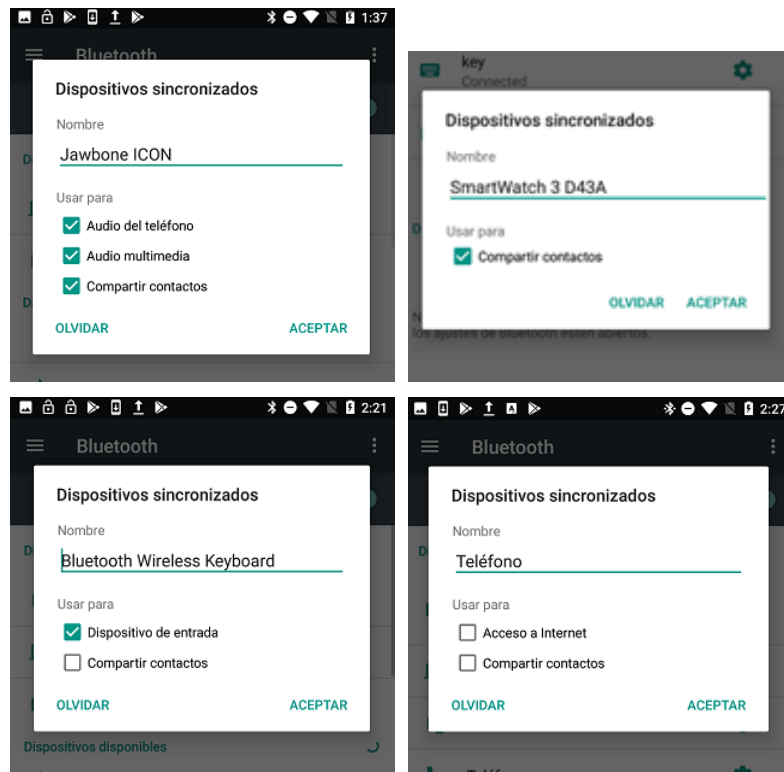


1510. Una vez se ha realizado la vinculación o emparejamiento con otro dispositivo, éste será mostrado bajo la sección "Dispositivos sincronizados" de la pantalla de configuración de Bluetooth. Además, se mostrará un icono descriptivo de la función (o clase Bluetooth) del dispositivo con el que se ha realizado el emparejamiento.
1511. No se han identificado opciones en Android para poder consultar y obtener la dirección Bluetooth (BD_ADDR) de cada dispositivo con el que se ha realizado el emparejamiento (o que ha sido descubierto), muy útil a la hora de reconocer (y confirmar) otros dispositivos asociados al terminal.
1512. Si un dispositivo Bluetooth con el que se ha realizado un emparejamiento se elimina del dispositivo móvil mediante la opción "Olvidar" y es posteriormente redescubierto, aunque dicho dispositivo haya cambiado su nombre Bluetooth, se redescubrirá con el último nombre que haya tenido mientras duró el emparejamiento previo (un comportamiento inesperado de Android 7.x).
1513. La información de la dirección sólo es mostrada temporalmente durante el proceso inicial de descubrimiento, hasta que es reemplazada por el

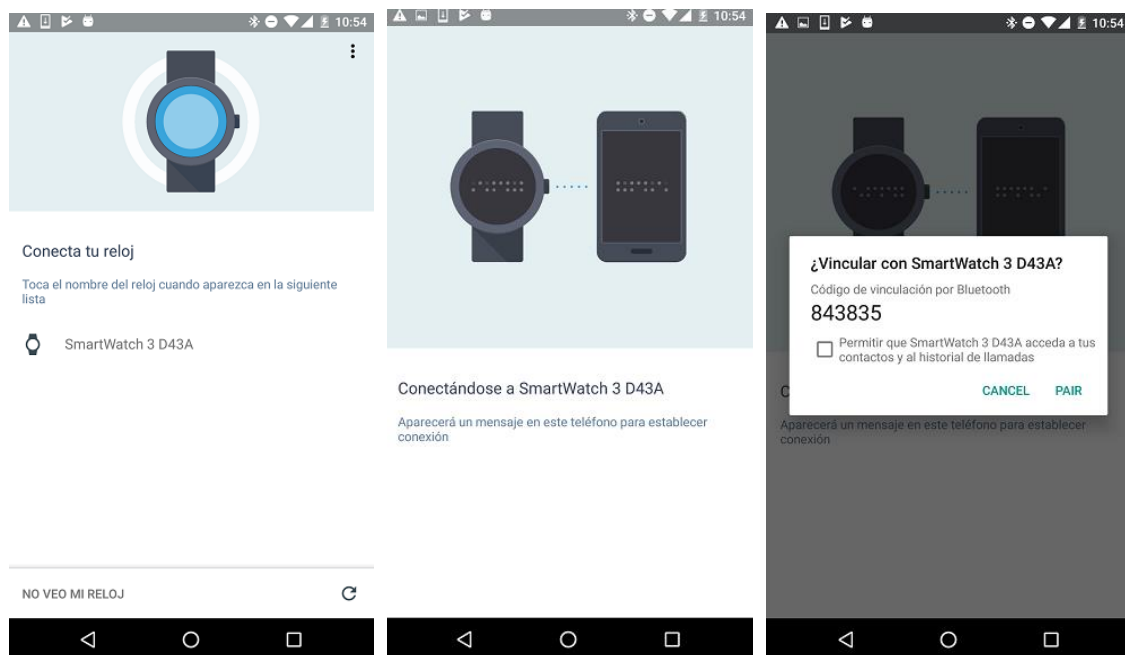
nombre Bluetooth del dispositivo correspondiente. En caso de no poder obtener el nombre, sí se seguirá mostrando la dirección Bluetooth.



1514. El icono situado a la izquierda de cada dispositivo permite conocer su tipo (en base a su clase Bluetooth). Adicionalmente, mediante la selección de los ajustes de configuración (icono con un engranaje) disponible a la derecha del nombre Bluetooth de cada dispositivo emparejado, es posible visualizar los posibles usos proporcionados por cada dispositivo, en función de los perfiles Bluetooth que declaran durante el proceso de descubrimiento:



1515. Asimismo, las opciones de configuración del dispositivo sincronizado permiten modificar el nombre empleado para identificarlo dentro de la lista de dispositivos emparejados con el dispositivo móvil, pudiendo así determinar más fácilmente si las conexiones con el dispositivo móvil deben ser autorizadas.
1516. Para la conexión de Android con un *smartwatch* (especialmente los basados en Android Wear), se requiere la instalación de la app "Android Wear" a través de Google Play, y realizar el emparejamiento a través de ella (a modo de ejemplo, se muestran sólo algunas imágenes relativas a este proceso, ya que la configuración de *smartwatches* queda fuera del alcance de la presente guía):



15.4. CONFIGURACIÓN AVANZADA DE BLUETOOTH

1517. La implementación de Bluetooth de Android, por defecto incluye diferentes perfiles (especificaciones de la funcionalidad y servicios proporcionados por los dispositivos Bluetooth), que dependen de la versión y dispositivo móvil utilizados.
1518. Algunos de los perfiles Bluetooth disponibles para el dispositivo móvil empleado para la elaboración de la presente guía son: "Headset Gateway" (para el envío de audio), "Handsfree Gateway" (perfil para dispositivos tipo manos libres), "AV Remote Control Target" (para control remoto de audio y vídeo), "Advanced Audio" (streaming de audio), "Personal Area Networking" (redes de área personal), "Android Network User", "MAP SMS" (Message Access Profile)⁹⁶, "OBEX Phonebook Access Server" (acceso a contactos) y "OBEX Push" (transferencia de objetos). La lista completa de los servicios

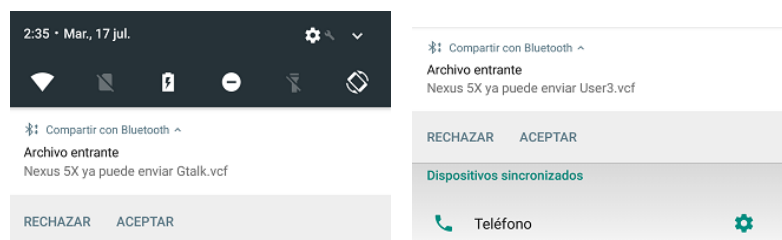
⁹⁶ El perfil de Bluetooth MAP permite al dispositivo móvil intercambiar mensajes con vehículos equipados con tecnología Bluetooth.

Bluetooth se puede consultar en la documentación oficial de Bluetooth para Android [Ref.- 632].

1519. No se han identificado en Android opciones de configuración avanzadas que permitan establecer restricciones de conectividad (autenticación, autorización y/o cifrado), obtener información, o configurar opciones específicas, para los diferentes perfiles Bluetooth disponibles (como por ejemplo para la configuración de red, de audio, de transferencia de archivos, etc.).
1520. La opción "Ajustes - [Conexiones inalámbricas y redes] Restablecer ajustes de red" provocará que se restablezcan, entre otros, todos los ajustes asociados a Bluetooth.

15.5. OBEX PUSH

1521. Android dispone por defecto de capacidades para el intercambio de objetos, asociadas al perfil Bluetooth estándar OBEX Push (en ocasiones referenciado como OBEX Object Push).
1522. El perfil de recepción de objetos (OBEX - *Object Exchange* - Push) a través de Bluetooth está habilitado por defecto y no se ha identificado ninguna opción de configuración para deshabilitarlo.
1523. Cuando otro dispositivo Bluetooth se comunica con este perfil e intenta enviar un objeto (por ejemplo, un archivo), Android solicita autorización por parte del usuario a través de una notificación emergente en la barra superior de estado antes de permitir la operación de intercambio. Se mostrará un símbolo de Bluetooth con par de puntos a su derecha en la parte superior izquierda de la barra de estado (en la versión 6.x de Android, se informaba de la solicitud de transferencia a través de un mensaje en la pantalla activa, pero este mensaje ya no se muestra en Android 7.x):



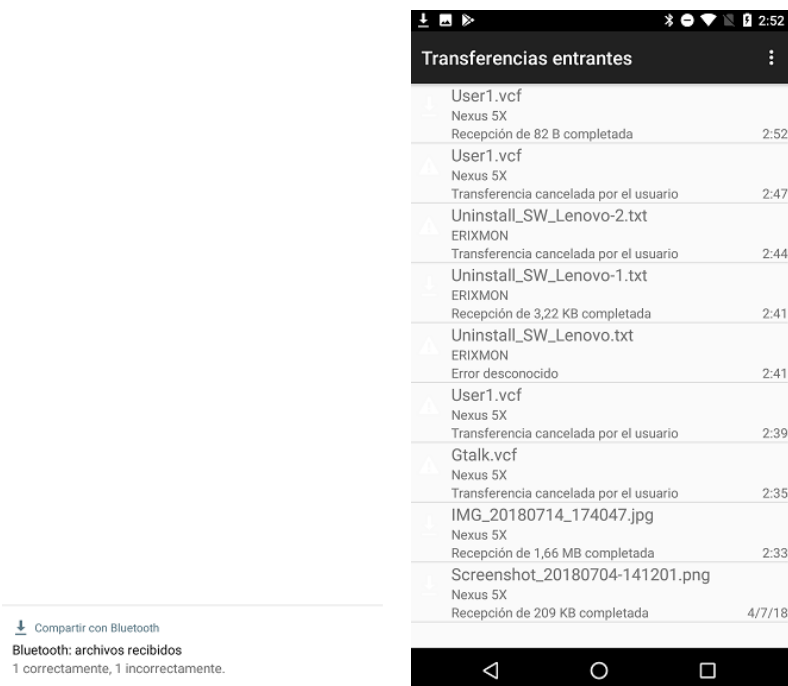
Nota: En la versión de Android 7.1.2 utilizada para la elaboración del presente apartado, el nombre que se mostrará en la notificación para el dispositivo emisor no se corresponde con el nombre que se haya configurado para él dentro de los ajustes de Bluetooth del dispositivo móvil, sino aquel con el que fue descubierto durante el emparejamiento inicial. Esto puede suponer un problema desde el punto de vista de seguridad, ya que, si se ha definido un nombre diferente para el dispositivo emisor, no se podrá establecer la correspondencia con el configurado actualmente: un potencial atacante podría enviar archivos suplantando al emisor si conoce el nombre que éste tuvo asignado durante el emparejamiento inicial. Este comportamiento se ilustra en la imagen superior derecha, en la cual se observa que el nombre inicial del dispositivo

emparejado, "Nexus 5X", se emplea en la notificación pese a haber sido cambiado a "Teléfono".

1524. Si se pulsa "Aceptar" en la notificación, comenzará la transferencia Bluetooth, y el progreso de la misma se mostrará a través de otra notificación:

Compartir con Bluetooth
Bluetooth: recibiendo /storage/emulated/0/bluetooth/IMG_...

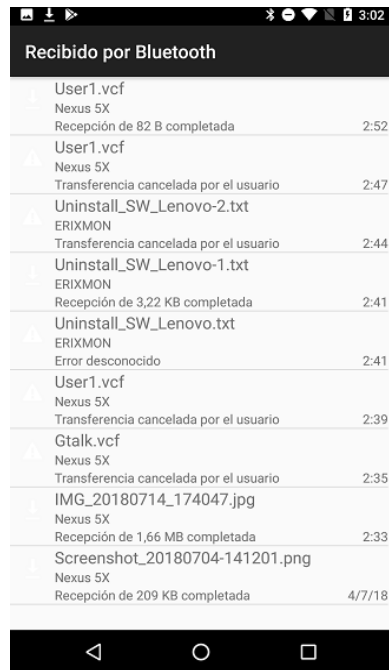
1525. Si la transferencia de datos a través del perfil OBEX Push tiene éxito, se generará una nueva notificación representada por una flecha apuntando hacia abajo (transferencia entrante) en la que se mostrará la estadística de transferencias. Desde esta notificación es posible acceder a la lista de archivos recibidos y, pulsando sobre cualquiera de ellos, abrirlos con la app encargada de su manejo según su tipo:



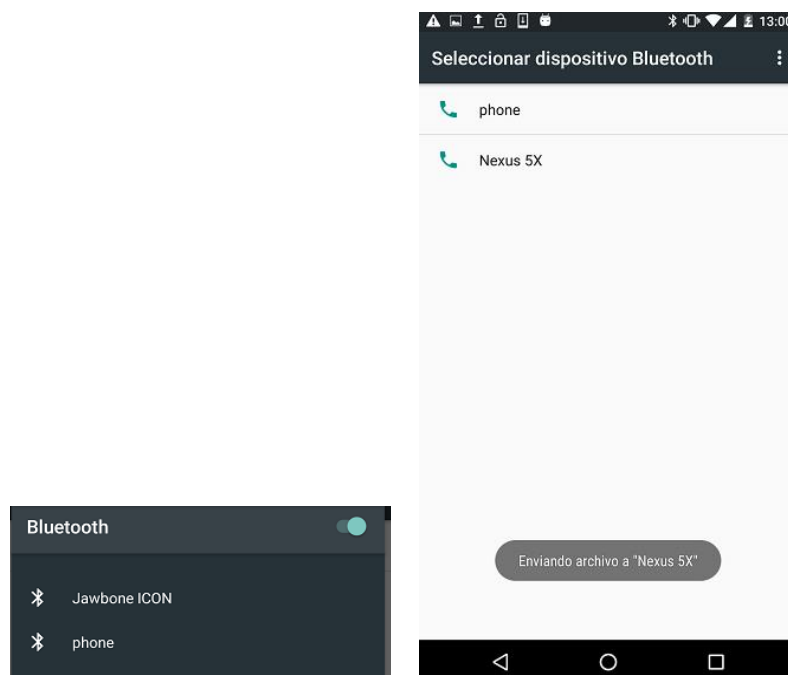
1526. El objeto (o fichero) recibido se almacena por defecto en el directorio "bluetooth" del área de almacenamiento interno del dispositivo móvil (ej. "/sdcard/bluetooth/"):

```
bullhead:/sdcard/bluetooth $ ls -xla
total 3432
drwxrwx--x 2 root sdcard_rw    4096 2018-07-17 02:52 .
drwxrwx--x 4 root sdcard_rw    4096 2018-07-04 14:17 ..
-rw-rw---- 1 root sdcard_rw 1738060 2018-07-17 02:34 IMG_20180714_174047.jpg
-rw-rw---- 1 root sdcard_rw    3293 2018-07-17 02:42 Uninstall_SW_Lenovo-1.txt
-rw-rw---- 1 root sdcard_rw      0 2018-07-17 02:41 Uninstall_SW_Lenovo.txt
-rw-rw---- 1 root sdcard_rw     82 2018-07-17 02:52 User1.vcf
```

1527. El acceso a los archivos recibidos mediante Bluetooth está disponible desde "Ajustes - Bluetooth - [...] - Mostrar archivos recibidos", incluyendo tanto los que han tenido éxito como los intentos fallidos, cancelados o rechazados por el usuario:



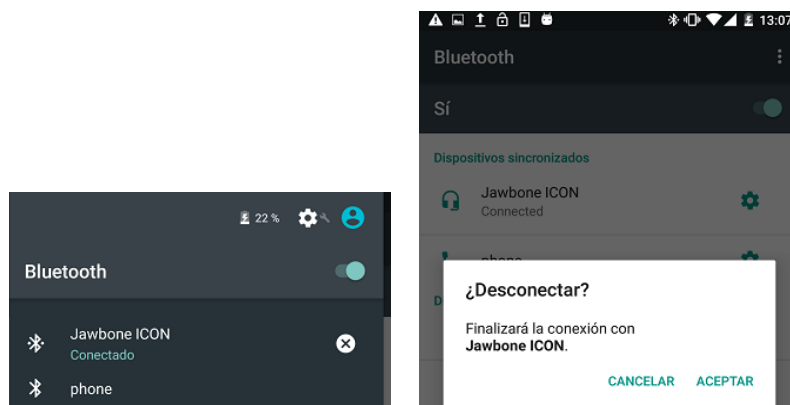
1528. El perfil OBEX Push no dispone del mecanismo de autenticación habilitado por defecto, debido a su modelo de uso, en el que no es necesario emparejarse previamente con el otro dispositivo para transferir datos (u objetos, OBEX), tal como ilustran las siguientes imágenes (la inferior izquierda muestra los dispositivos vinculados, la derecha los dispositivos con los que se puede compartir un archivo mediante OBEX Push):



1529. El comportamiento de solicitud de autorización al usuario descrito previamente se presenta tanto con dispositivos Bluetooth con los que previamente se ha emparejado Android (es decir, autenticados), como con dispositivos nuevos.

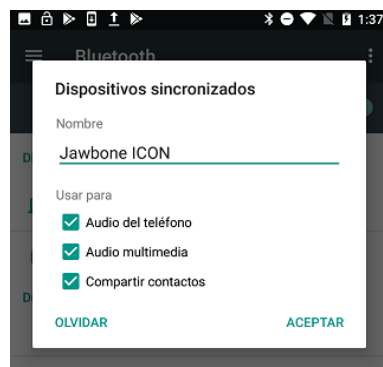
15.6. MANOS LIBRES BLUETOOTH

1530. El emparejamiento con dispositivos Bluetooth manos libres puede no requerir el introducir ningún PIN o contraseña, proceso conocido como emparejamiento automático o *auto pairing* (método "Just Works" de la especificación Bluetooth 2.1), debido a que Android dispone de conocimiento de los PINs empleados comúnmente por defecto (no modificables) por los fabricantes de manos libres Bluetooth.
1531. El emparejamiento automático se lleva a cabo por ejemplo con dispositivos que emplean valores de PIN por defecto, como por ejemplo "0000".
1532. Una vez vinculado el dispositivo móvil con un dispositivo manos libres, el interfaz Bluetooth del dispositivo móvil, cada vez que es habilitado, intenta establecer automáticamente una conexión con los dispositivos manos libres disponibles en la lista de dispositivos emparejados, proceso también conocido como conexión automática o *auto connect*. Si el dispositivo Bluetooth se encuentra en el rango de alcance del dispositivo móvil, y fueron emparejados previamente, se establecerá la conexión, estado identificado en Android por el icono de estado de Bluetooth en la barra superior de estado.
1533. El proceso de conexión automática no tiene asociada ninguna solicitud de confirmación por parte del usuario ni puede ser deshabilitado mediante opciones de configuración en Android (salvo deshabilitando el interfaz Bluetooth).
1534. Una vez se ha establecido la conexión Bluetooth con otro dispositivo, es posible desconectarse del mismo bien a través del símbolo "x" del menú de ajustes rápidos, bien al seleccionarlo a través del menú de ajustes de Bluetooth y pulsando el botón "Aceptar":



1535. Una vez se ha establecido la conexión Bluetooth con otro dispositivo (independientemente de su tipo o del perfil empleado), al seleccionar su botón de configuración asociado (disponible en la parte derecha, junto a su nombre), se dispone de un menú de opciones. A través del mismo es posible modificar el nombre empleado localmente para identificar al otro dispositivo, eliminar la vinculación o emparejamiento previos mediante el botón

"Olvidar", o seleccionar los perfiles ofrecidos por el otro dispositivo que se desean utilizar:



1536. No se han identificado en Android opciones de configuración avanzadas que permitan configurar los perfiles de Bluetooth asociados a la conexión con dispositivos manos libres.

15.7. CONEXIÓN DE RED COMPARTIDA POR BLUETOOTH

1537. Android proporciona capacidades nativas de *tethering* a través de USB (ver apartado "13.2. Conexión de red compartida por USB (*Tethering*)").
1538. Para hacer uso de estas capacidades mediante Bluetooth (Personal Area Network, PAN) es necesario acceder al menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Compartir Internet y zona Wi-Fi", y en concreto a la opción "Compartir por Bluetooth":



1539. A través de la opción "Compartir por Bluetooth" es posible habilitar el servicio de *tethering*, de forma que el ordenador utilice el dispositivo móvil como su conexión de red a través de una conexión Bluetooth. Para ello es necesario que previamente el ordenador y el dispositivo móvil se hayan emparejado.
1540. La configuración empleada para Bluetooth es similar a la utilizada para compartir la conexión mediante USB (ver apartado "13.2. Conexión de red compartida por USB (*Tethering*)") o Wi-Fi (ver apartado "16.6. Punto de acceso o zona Wi-Fi").
1541. Tras establecerse la conexión de red, y tomando Windows 10 como sistema operativo del ordenador de referencia, se crea automáticamente un interfaz de red de tipo "Bluetooth Device (Personal Area Network)" (PAN) configurado con una dirección IP dinámica (DHCP) por defecto de clase C (/24), como por ejemplo 192.168.44.191.

```

Adaptador de Ethernet Conexión de red Bluetooth:
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::159a:41a8:e3f8:dfaf%14
Dirección IPv4. . . . . : 192.168.44.191
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.44.1

```

1542. Este interfaz de red Bluetooth del ordenador tiene asociada la dirección MAC del adaptador Bluetooth empleado en el ordenador.

1543. La configuración de este interfaz de red emplea por defecto DHCP (a través del servidor DHCP de Android), y toma todos los valores por defecto existentes en el sistema operativo del ordenador. Así por ejemplo en Windows 10 la pila de comunicaciones de IP versión 6 está activa, al igual que NetBIOS sobre TCP/IP.

1544. Se recomienda modificar y restringir la configuración de este interfaz de red para habilitar únicamente los protocolos y servicios que se consideren necesarios en el ordenador.

1545. El dispositivo móvil, con dirección IP 192.168.44.1, actúa de gateway, router o pasarela por defecto, de servidor DHCP, y de servidor DNS, y es alcanzable mediante ping (ICMP).

```

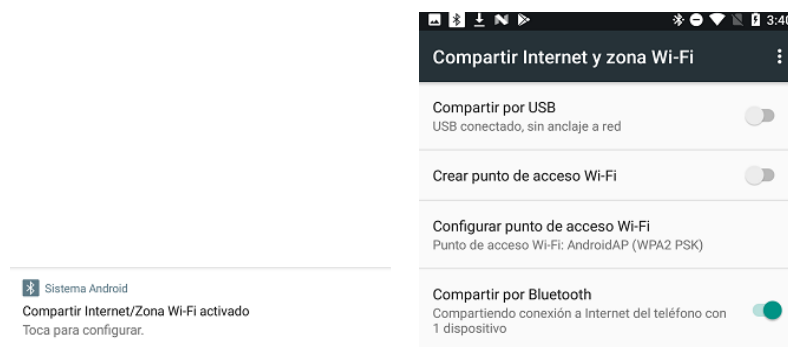
shell@bullhead:/ $ ifconfig
bt-pan  Link encap:UNSPEC
        inet addr:192.168.44.1  Bcast:192.168.44.255  Mask:255.255.255.0
        inet6 addr: fe80::825a:4ff:fe0e:15aa/64 Scope: Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:54 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:500
        RX bytes:5105 TX bytes:318

```

Nota: El direccionamiento IP empleado por Android para la conexión de red compartida a través de Bluetooth (192.168.44.x/24) no puede ser modificado salvo en dispositivos móviles *rooteados* [Ref.- 99].

1546. La dirección MAC asociada al interfaz de red empleado para proporcionar la conexión de red compartida a través de USB por el dispositivo móvil tiene asociada la dirección MAC del interfaz Bluetooth del dispositivo móvil.

1547. El dispositivo móvil informa de la compartición de la conexión Bluetooth a través de una notificación (accesible desde la barra superior de notificaciones):



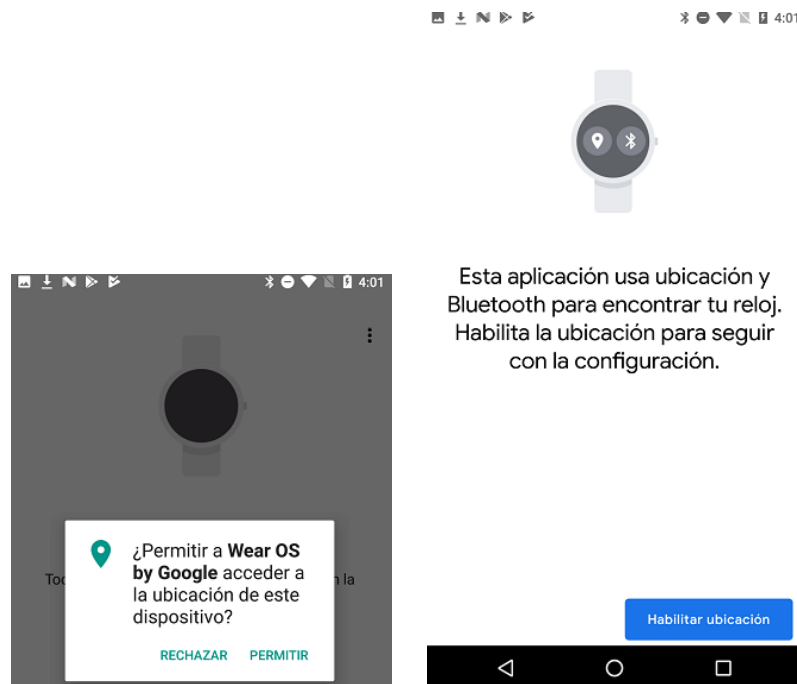
1548. Es posible finalizar la compartición de la conexión de datos deshabilitando la opción "Compartir por Bluetooth" habilitada previamente, opción recomendada desde el punto de vista de seguridad una vez no se desea disponer de la conexión de red compartida.

15.8. BLUETOOTH LOW ENERGY (BLE)

1549. Android 5.x añadió capacidades de Bluetooth Low Energy (BLE) para que el dispositivo móvil pueda actuar como un periférico (realizando anuncios, o actuando en modo periférico de BLE), en lugar de únicamente como el elemento central, funcionalidad introducida en Android 4.3 (realizando escaneos, o actuando en modo central de BLE).
1550. Las apps pueden hacer uso de esta nueva funcionalidad para anunciar su presencia de cara a otros dispositivos cercanos. Así, una app puede actuar como un sensor o monitor de salud y proporcionar los datos recolectados a otro dispositivo BLE.
1551. Para ello las apps disponen de la posibilidad de enviar anuncios de difusión (*broadcast advertisements*), escanear las respuestas recibidas y establecer las conexiones asociadas. Esta funcionalidad requiere que la app declare el permiso `BLUETOOTH_ADMIN`, que se considera un permiso estándar o normal (ver apartado "7.1.3. Permisos normales y peligrosos"), por lo que no se solicitará explícitamente su concesión al usuario en tiempo de instalación y/o ejecución si la app lo declara en su fichero *manifest*, ni se podrá habilitar/deshabilitar de forma independiente:



1552. Asimismo, Android 5.x introdujo capacidades BLE para que una app pueda aplicar filtros durante el proceso de escaneo y sólo descubra aquellos dispositivos en los que está interesada.
1553. En Android, un dispositivo BLE puede ser accedido por más de una app al mismo tiempo. La conexión física la gestiona el sistema operativo, y cada app accede a ella como si tuviese la conexión en exclusiva.
1554. Android 6.x añadió mejoras para las apps que hacen escaneos BLE, pudiendo recibir notificaciones del sistema tan pronto se reciba un anuncio BLE vinculado al filtro de escaneo especificado, mejorando así el consumo energético asociado a este tipo de tareas [Ref.- 603].
1555. Cuando se establece una conexión con un dispositivo BLE, la app debe pasar un parámetro, *autoconnect*, que determina si Android debería reintentar la conexión en caso de que ésta se vea interrumpida (salvo que la interrupción se deba a la inhabilitación del interfaz Bluetooth o a que la app solicite la desconexión deliberadamente). Éste es el caso de la conexión con un *smartwatch*, de forma que, si éste sale del rango de alcance, cuando vuelva a entrar en él, se restablezca la conexión.
1556. Cuando el parámetro *autoconnect* tiene valor "verdadero", no se establecerá un temporizador para los intentos de conexión. Si tiene valor "falso", se establecerá un temporizador de 30 segundos para la conexión.
1557. Para permitir el escaneo BLE, en Android 6.x se requiere habilitar el servicio de ubicación y conceder a la app permiso de localización (este requisito no existía en versiones anteriores de Android). Este requisito (permiso considerado peligroso por Google) se solicita al usuario en tiempo de ejecución desde Android 6.x.

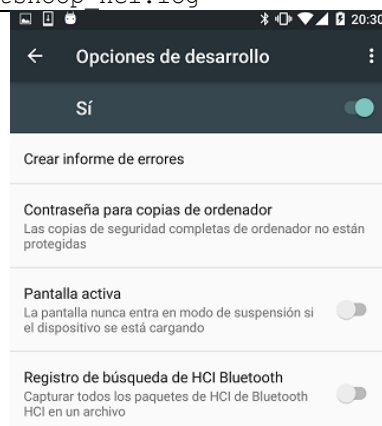


1558. Android 7.x introdujo un cambio en el escaneo BLE para protegerse contra ataques conocidos como "BLE Scanning Abuse" [Ref.- 746], de forma que:

- Una app no podrá iniciar/detener más de 5 escaneos en un periodo de 30 segundos.
- Una app no podrá realizar escaneos demasiado largos (más de 30 minutos). Los escaneos que superen este umbral se convertirán en "escaneos oportunistas"⁹⁷, que sólo recibirán resultados si un escaneo no oportunista los produce en base al mismo filtro.

1559. Existe una opción de desarrollo que permite rastrear las conexiones Bluetooth establecidas por el dispositivo móvil. Esta opción, que se ilustra en la imagen siguiente, al activarse provoca que se cree un fichero de log en la ubicación especificada por la variable de configuración correspondiente del fichero "/etc/bluetooth/bt_stack.conf":

```
shell@bullhead:/ $ cat /etc/bluetooth/bt_stack.conf | grep BtSnoopFileName
BtSnoopFileName=/sdcard/btsnoop_hci.log
```



⁹⁷ <https://blog.classycode.com/undocumented-android-7-ble-behavior-changes-d1a9bd87d983>

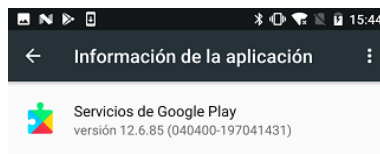
1560. El fichero "btsnoop_hci.log" puede leerse con Wireshark [Ref.- 257], y en el mismo se registra toda la actividad tanto entre el dispositivo móvil y los dispositivos Bluetooth externos, como entre el denominado "controller" (hardware local responsable del procesamiento de tráfico Bluetooth) y el denominado "host" (la pila software Bluetooth que implementa el propio sistema operativo Android).

15.9. GOOGLE NEARBY

1561. Google Nearby se lanzó como un servicio de búsqueda que, en función de la ubicación del dispositivo móvil, proporcionaba al usuario información de contexto sin requerir disponer de una app específica que la gestionase.
1562. En sus orígenes, se fundamentaba en las denominadas "Nearby Notifications", soportadas desde Android 4.4 (KitKat), mediante la generación de notificaciones dependientes de la ubicación en apps y sitios webs. Por ejemplo, estas permiten mostrar mensajes sobre productos en promoción dentro de un supermercado, guiar al usuario dentro de un museo, o iniciar conversaciones entre apps de mensajería.
1563. Para ello, la app o sitio web que desea hacer uso de estas notificaciones, debe asociarse con un dispositivo BLE (Bluetooth LE o Smart) o un beacon EddyStone o iBeacon [Ref.- 254]. Cuando un dispositivo móvil Android con capacidad Nearby se aproxima a un dispositivo de ese tipo, la pizarra de ajustes "Nearby Quick Settings" parpadeará y se generarán mensajes en forma de notificación que permiten dirigir al usuario a la Play Store, a un sitio móvil web o directamente a una funcionalidad específica de una app instalada.
1564. La funcionalidad de notificaciones Nearby está implementada internamente en los servicios de Google Play.
1565. Las denominadas "Nearby Connections" son una API de comunicaciones extremo a extremo destinada al descubrimiento, conexión e intercambio de datos entre dispositivos en tiempo real, aunque no estén conectados a través de una red con una infraestructura [Ref.- 281] [Ref.- 282].
1566. Para ello, las conexiones Nearby utilizan hotspots Wi-Fi, y comunicaciones BLE y Bluetooth para establecer las comunicaciones entre dispositivos con capacidad Nearby. En función de la intensidad y calidad de cada una de dichas señales inalámbricas, se cambiará de una a otra.
1567. En el Google I/O de septiembre de 2017, Google comunicó las novedades relativas a la API de conexiones Nearby, que proporcionan mayor ancho de banda, menor latencia y transferencias de datos cifradas entre dispositivos con capacidades Nearby, y basadas en comunicaciones P2P offline⁹⁸. Estas novedades están disponibles desde la versión 11.0 de los servicios de Google Play.

⁹⁸ <https://youtu.be/1a0wll96cpE?t=23m2s>

1568. La versión de los servicios de Google Play es la 12.6.85 a fecha de elaboración de la presente guía.



1569. Su configuración detallada se realiza a través del menú "Ajustes - [Personal] Google - Nearby", pero queda fuera del alcance de la presente guía, por lo que, en caso de desear utilizarse, se recomienda consultar la ayuda de Google [Ref.- 274].
1570. No obstante, debido al uso que hace de las comunicaciones Bluetooth y Wi-Fi, la recomendación desde el punto de vista de seguridad sería poder deshabilitar el servicio Nearby.

Importante: No se conoce ningún ajuste en Android 7.x que permita deshabilitar el uso de Nearby en los dispositivos Nexus o Pixel. Ni tan siquiera mantener deshabilitados los interfaces Bluetooth o Wi-Fi impedirá que se realicen escaneos Bluetooth y/o Wi-Fi si en el sistema existen aplicaciones con capacidades Nearby, tal como se constata en la siguiente imagen, procedente de la ayuda de Google para Nearby [Ref.- 274].

Sí se tiene constancia de que otros fabricantes ofrecen ajustes para deshabilitar estos escaneos.

Comprobar los ajustes y la conectividad

Comprobar los ajustes de ubicación y de Bluetooth

Si el Bluetooth o la ubicación están desactivados, no recibirás listas ni notificaciones Nearby.

Si concedes permiso a una aplicación para utilizar Nearby y el Bluetooth está desactivado, Nearby activará el Bluetooth de forma temporal para establecer la conexión.

Más información sobre cómo comprobar los ajustes de [ubicación](#) y de [Bluetooth](#)

Comprobar la conexión móvil o Wi-Fi

Si tu dispositivo no está conectado a Internet, no recibirás listas ni notificaciones Nearby.

Si concedes permiso a una aplicación para utilizar Nearby y la conexión Wi-Fi está desactivada, Nearby activará la conexión Wi-Fi de forma temporal para establecer la conexión.

Más información sobre cómo comprobar la conexión a la [red Wi-Fi](#) o la [red móvil](#)

1571. En general, desde el punto de vista de seguridad, se recomienda prescindir de las aplicaciones que hagan uso de ese servicio.

16. COMUNICACIONES WI-FI

16.1. UTILIZACIÓN Y CONFIGURACIÓN DE LAS COMUNICACIONES WI-FI

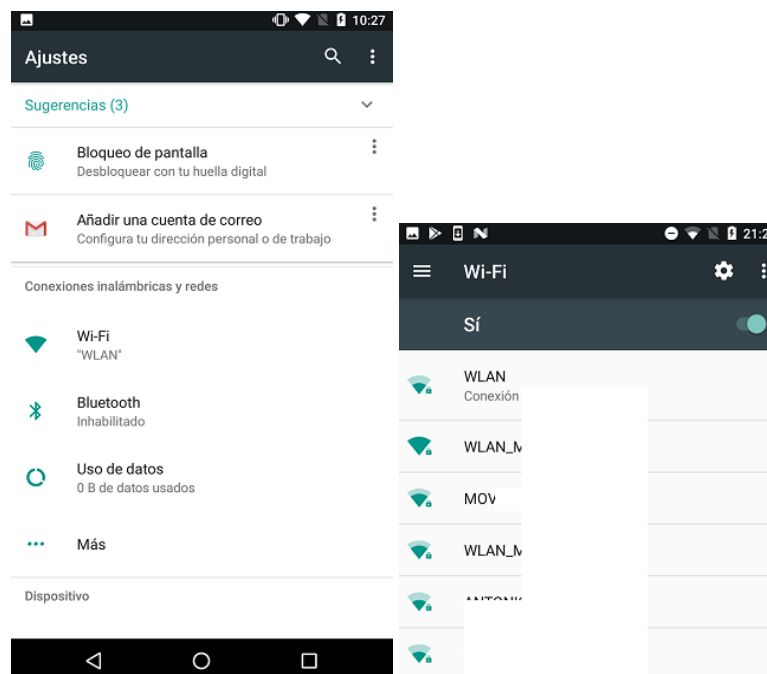
1572. La principal recomendación de seguridad asociada a las comunicaciones Wi-Fi (802.11) en dispositivos móviles es no activar el interfaz inalámbrico Wi-Fi salvo en el caso en el que se esté haciendo uso del mismo, evitando así

la posibilidad de ataques sobre el hardware del interfaz, el driver o la pila de comunicaciones Wi-Fi [Ref.- 48].

Nota: Durante el proceso de configuración inicial del dispositivo móvil empleado para la elaboración de la presente guía (que se inicia automáticamente la primera vez que se enciende el terminal, según se ilustra en el apartado "23. Apéndice A: Proceso de instalación y configuración inicial"), tanto si se omite el establecimiento de una conexión a una red Wi-Fi como si no, el interfaz Wi-Fi estará habilitado por defecto.

Desde el punto de vista de seguridad se recomienda deshabilitar el interfaz Wi-Fi y habilitarlo posteriormente, una vez el usuario se encuentre en un entorno seguro y de confianza para poder establecer una conexión a una red Wi-Fi.

1573. El interfaz inalámbrico Wi-Fi puede ser desactivado en Android a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Wi-Fi"⁹⁹, mediante el botón "SÍ|NO" de la opción "Wi-Fi":



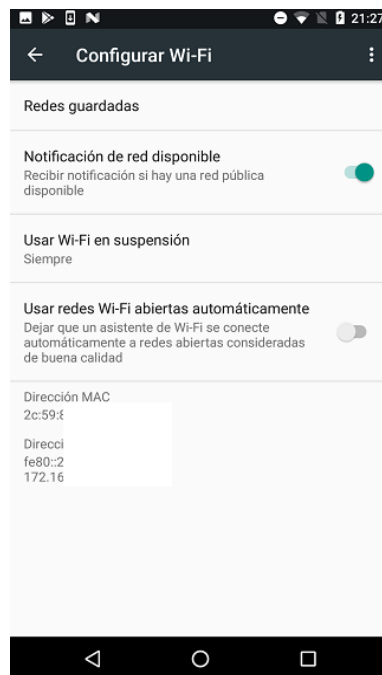
1574. Android proporciona información en la barra superior de estado sobre la disponibilidad de redes Wi-Fi abiertas y si se está o no conectado a una red Wi-Fi actualmente, a través de un conjunto específico de iconos que muestran las ondas de las señales Wi-Fi.

1575. La disponibilidad de redes Wi-Fi abiertas se muestra con un símbolo de interrogación en las ondas de las señales Wi-Fi, en la zona izquierda de la barra superior de estado (correspondiente a las notificaciones), acompañado de un nuevo mensaje de notificación:



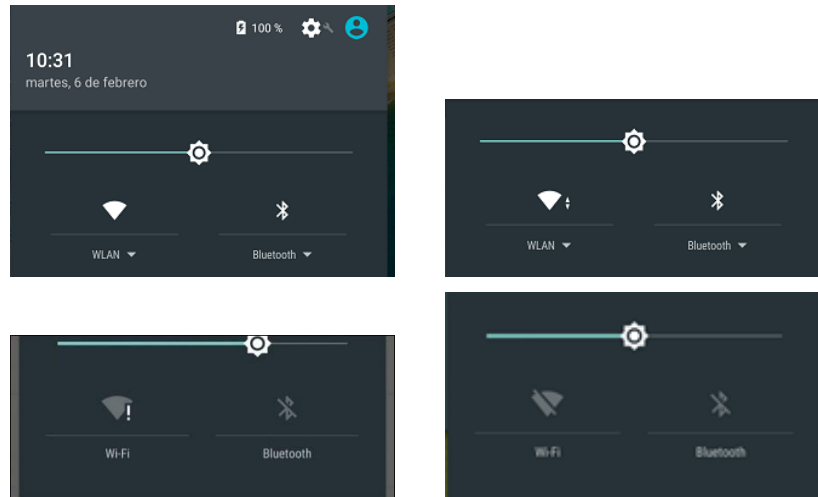
⁹⁹ A partir de este momento, por simplificar, el menú "Ajustes [Conexiones Inalámbricas y Redes] - Wi-Fi" se abreviará como "Ajustes - Wi-Fi".

1576. La notificación respecto a la disponibilidad de redes Wi-Fi abiertas puede ser configurada a través del menú "Ajustes - Wi-Fi - [icono de engranaje] - Notificación de red disponible" (habilitada por defecto). Esta opción puede ser modificada únicamente una vez el interfaz Wi-Fi ha sido habilitado.
1577. Adicionalmente, se ofrece la opción "Usar redes Wi-Fi abiertas automáticamente", que, estando activa, provoca que el dispositivo móvil se conecte de forma automática a redes Wi-Fi abiertas. Esta opción se considera totalmente desaconsejada desde el punto de vista de seguridad, ya que a través suyo el propietario del dispositivo pierde el control sobre las redes Wi-Fi con las que se establecerá conexión:

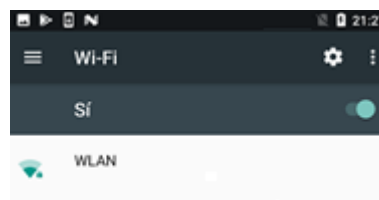


1578. Dado que desde el punto de vista de seguridad se desaconseja el establecer conexiones con redes Wi-Fi abiertas, se recomienda desactivar también la opción de "Notificación de red disponible".
1579. Los ajustes rápidos, disponibles al desplazar hacia abajo la barra de menú superior, permiten conocer más detalles sobre el interfaz Wi-Fi. Por ejemplo, si las ondas Wi-Fi aparecen con una admiración, ésta puede indicar que se han detectado problemas de conectividad hacia Internet¹⁰⁰; un icono atravesado por una línea diagonal indica que el interfaz Wi-Fi está desactivado.

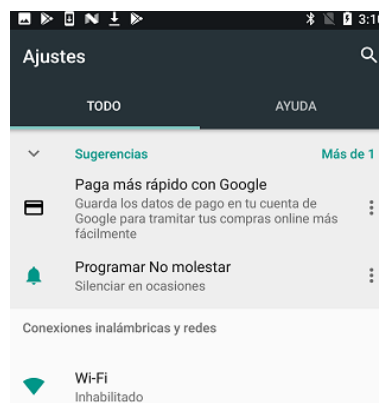
¹⁰⁰ En algunas versiones de Android, la existencia de flechas hacia arriba y hacia abajo en el icono Wi-Fi indican que el dispositivo móvil está transmitiendo o recibiendo datos, respectivamente.



1580. Desafortunadamente, la barra superior de estado no proporciona información sobre el estado de activación del interfaz Wi-Fi, ya que únicamente se mostrará el icono correspondiente si el dispositivo móvil está conectado a una red Wi-Fi actualmente, y no lo mostrará si no lo está aunque el interfaz se encuentre habilitado:



1581. Sin embargo, a diferencia de lo que sucedía en Android 6.x, la pantalla principal de "Ajustes" sí permite ver el estado actual del interfaz Wi-Fi:



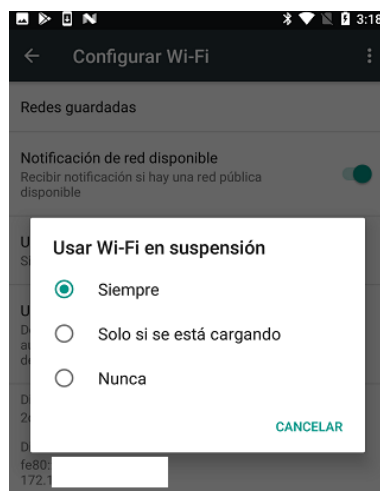
1582. Android mantiene el estado del interfaz Wi-Fi tras reiniciar el dispositivo móvil, es decir, si el interfaz Wi-Fi estaba activo al apagar el terminal, al encender el dispositivo móvil seguirá activo, incluso con el terminal bloqueado.
1583. Igualmente, al salir del modo avión el estado del interfaz Wi-Fi será restaurado, activándose de nuevo en el caso de haber estado activo previamente (antes de activar el modo avión).
1584. Cuando el dispositivo móvil está conectado a una red, el interfaz Wi-Fi permanece activo tras pasar al estado de espera o *stand-by* (desatendido), es decir, el estado en que el dispositivo móvil está encendido pero con la

pantalla apagada y bloqueado (ver posteriormente la opción avanzada de configuración asociada), y sigue generando tráfico periódicamente.

1585. En caso de no estar conectado a una red Wi-Fi, en el estado de espera (o suspendido) el dispositivo móvil genera por defecto el tráfico 802.11 asociado a las tramas de verificación de disponibilidad de redes Wi-Fi (*probe request*).

Nota: Por ejemplo, inicialmente cada 10 segundos por defecto, y mientras el dispositivo móvil no está conectado a ninguna red Wi-Fi, el dispositivo móvil genera tramas *probe request* de tipo *broadcast* para verificar la disponibilidad de redes Wi-Fi en su zona de alcance. Posteriormente, este intervalo puede ser mayor, de unos 15 segundos aproximadamente, y en ocasiones superándose este valor, por ejemplo, hasta 20 segundos; la frecuencia de escaneo es dependiente de la versión de Android y del estado del interfaz Wi-Fi.

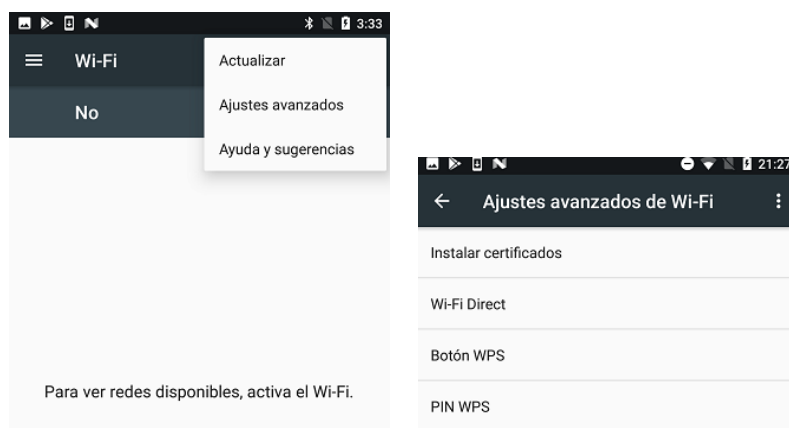
1586. El comportamiento del interfaz Wi-Fi respecto a las operaciones de ahorro de energía del dispositivo móvil puede ser configurado desde el menú "Ajustes - Wi-Fi - [icono de engranaje]", a través de las siguientes opciones:
1587. La opción "Usar Wi-Fi en suspensión" (que puede modificarse incluso aunque el interfaz esté deshabilitado) permite acceder a las diferentes alternativas de configuración disponibles:



- "Siempre" (opción por defecto): permite que el interfaz Wi-Fi esté activo y sea utilizado siempre, incluso cuando el dispositivo móvil está suspendido.
 - "Solo si se está cargando": permite el uso de Wi-Fi con el dispositivo móvil suspendido sólo si está conectado a la electricidad. En caso de estar alimentado solo por la batería, entrarán en vigor las consideraciones del modo Doze (ver apartado "9.9. Ahorro de batería (Doze)"), que afectan, entre otros, a los escaneos Wi-Fi.
 - "Nunca": nunca se hace uso de Wi-Fi si el dispositivo móvil está suspendido, por lo que en ese caso se hará uso de la conexión de datos de telefonía móvil (aumentando potencialmente el consumo de datos).
1588. Las opciones "Notificación de red disponible" y "Usar redes Wi-Fi abiertas automáticamente" deben desmarcarse para evitar conexiones a

redes abiertas de forma no controlada (es preciso que el interfaz esté habilitado para poder modificarlas).

1589. Al igual que ocurría en Android 6.x y 5.x, en Android 7.x desaparecen las opciones "Optimización de Wi-Fi" y "Solo conexiones estables" (existentes ambas en versiones previas de Android hasta la 4.4.x).
1590. Una vez inhabilitado el interfaz Wi-Fi, el dispositivo móvil sólo hará uso de las capacidades de datos de telefonía móvil (ver apartado "18. Comunicaciones móviles: voz y datos").
1591. Además de las opciones accesibles a través del menú correspondiente al símbolo de engranaje, existen otras opciones disponibles en el menú "Ajustes - Wi-Fi - [...] Ajustes avanzados".



1592. Desaparece la opción "Banda de frecuencia Wi-Fi" presente en versiones anteriores y que permitía seleccionar entre una configuración automática (por defecto) o manual, eligiendo sólo la banda de 2,4 GHz o sólo la de 5 GHz.
1593. Debe tenerse en cuenta que, al igual que ocurría con el interfaz Bluetooth, para poder llevar a cabo la configuración completa de la pila de comunicaciones Wi-Fi a través del interfaz de usuario del dispositivo móvil es obligatorio activar el interfaz, por lo que se recomienda realizar las modificaciones de la configuración en un entorno seguro, evitando así potenciales ataques cuando los mecanismos de seguridad no han sido aún aplicados.
1594. La funcionalidad asociada a la opción para añadir una red Wi-Fi manualmente, mediante el menú "Añadir red" no está disponible (o habilitada) hasta que el interfaz Wi-Fi no ha sido activado.
1595. La dirección del interfaz Wi-Fi está disponible a través del menú "Ajustes - Información del teléfono - Estado", y en concreto, en el campo "Dirección MAC de Wi-Fi". La dirección es visible tanto si el interfaz Wi-Fi se encuentra activo como si está desactivado. También se dispone de la dirección en la parte inferior de la pantalla correspondiente al icono de engranaje:



Nota: La parte de la dirección Wi-Fi asociada al fabricante (primeros tres *bytes* de la dirección) puede permitir identificar el tipo de dispositivo móvil. En el ejemplo analizado, el valor "78:f8:82" está asignado por el IEEE (<http://standards.ieee.org/regauth/oui/index.shtml>) a "LG":

78-F8-82	(hex)	LG Electronics (Mobile Communications)
78F882	(base 16)	60-39, Gasan-dong, Geumcheon-gu
		Seoul Seoul 153-801
		KOREA, REPUBLIC OF

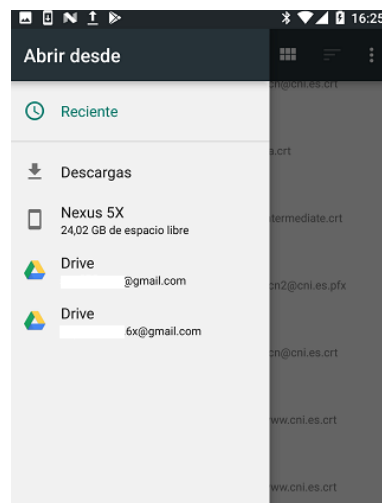
1596. Es aconsejable comparar la dirección Wi-Fi del dispositivo móvil con su dirección Bluetooth (ver apartado "15. Comunicaciones Bluetooth"), con el objetivo de identificar si se trata de direcciones correlativas. En caso afirmativo, un potencial atacante podría obtener la dirección de Bluetooth a partir de la dirección Wi-Fi, con los riesgos que ello conlleva [Ref.- 400], al emplearse la ocultación de la dirección Bluetooth en diferentes escenarios como mecanismo de seguridad.

1597. En el caso del dispositivo móvil empleado para la elaboración de la presente guía ambas direcciones comparten el mismo rango del fabricante, pero sin ningún otro elemento común (asignación aleatoria dentro de ese rango).

1598. Las opciones avanzadas incluyen:

- Instalar certificados: permite acceder directamente al almacenamiento interno del dispositivo móvil para instalar un certificado digital para su uso en redes Wi-Fi. Es importante destacar que los certificados instalados para Wi-Fi no se visualizan en el almacén de credenciales (ver apartado "11.3. Gestión de certificados digitales"), sino solo a través del menú de configuración de una red Wi-Fi de tipo empresarial (o EAP/802.1x).

Para instalar un certificado mediante esta opción, es necesario que el fichero asociado a él se pueda abrir desde almacenamiento interno, externo o a través de una cuenta de Google Drive configurada en el dispositivo móvil:



- Wi-Fi Direct: descrita en el apartado "16.7. Wi-Fi Direct ("P2P")".
- Botón WPS: permite iniciar una conexión con un punto de acceso Wi-Fi con soporte para WPS (Wi-Fi Protected Setup) en la modalidad "WPS Push", que no precisa definir ni un nombre de red ni una clave y que requiere pulsar el botón del punto de acceso Wi-Fi durante la conexión (lo cual requiere proximidad física al mismo).
- PIN WPS: este tipo de conexión se realiza a través de un código numérico o PIN (de 8 dígitos) que debe introducirse en el punto de acceso Wi-Fi para establecer la conexión WPS con el dispositivo móvil. Al pulsar esta opción, Android mostrará este PIN (que cambia de una vez a otra) .

Nota: El uso de WPS con PIN está desaconsejado desde el punto de vista de seguridad por ser vulnerable a ataques de fuerza bruta.

16.2. SELECCIÓN DE LA RED DE DATOS EN ANDROID

Nota: La selección de la red de datos en Android afecta principalmente a la red de datos de telefonía móvil 2/3/4G (analizada en el apartado "18. Comunicaciones móviles: voz y datos") y al interfaz Wi-Fi (ver el presente apartado).

1599. Debido a las múltiples opciones de conectividad de datos disponibles en los dispositivos móviles, como Bluetooth, y principalmente Wi-Fi y 2/3/4G, en caso de haber múltiples opciones habilitadas y disponibles, Android gestiona cual es la mejor conexión de datos a usar en cada momento a través del servicio Connectivity Service (o Connectivity Manager) [Ref.- 51]. De esta forma evita que el usuario tenga que decidir constantemente que conexión utilizar y proporciona la conectividad necesaria a las apps que ejecutan en el dispositivo móvil.

1600. El Connectivity Service emplea diferentes criterios para seleccionar y mantener la mejor y única opción de conexión de datos, que por defecto da prioridad al interfaz Wi-Fi frente al interfaz de telefonía móvil (2/3/4G):

- Si Wi-Fi está activo y se activa 2/3/4G, este segundo no llegará a ser utilizado.
- Si 2/3/4G está activo y se activa Wi-Fi, la conexión 2/3/4G dejará de ser utilizada y pasará a hacerse uso de la conexión Wi-Fi.
- Si Wi-Fi está activo y se desactiva, el servicio comprobará el estado y la configuración de 2/3/4G, y si está disponible, lo habilitará y hará uso de este interfaz de red.

1601. Los eventos de activación y desactivación de los dos interfaces, Wi-Fi y 2/3/4G, pueden ser visualizados a través de los iconos del menú superior de Android.

1602. El comportamiento del interfaz Wi-Fi, y en consecuencia del interfaz 2/3/4G, respecto a las operaciones de ahorro de energía del dispositivo móvil cuando éste se encuentra en estado suspendido, puede ser configurado desde el menú de ajustes de Wi-Fi, tal y como se detalla en el apartado "16.1. Utilización y configuración de las comunicaciones Wi-Fi".

1603. El Connectivity Service no permite disponer de varias conexiones activas simultáneamente, opción recomendada desde el punto de vista de seguridad para evitar escenarios *dual homing* donde la conexión Wi-Fi y la conexión de datos de telefonía móvil (2/3/4G) están activas simultáneamente, con el objetivo de que el dispositivo móvil no actúe de pasarela entre distintas redes.

1604. Desde el punto de vista de seguridad podría considerarse más seguro cursar todo el tráfico de datos a través de las redes de telefonía móvil (cuando estén disponibles según la cobertura existente) en lugar de a través de redes Wi-Fi (especialmente si se quiere hacer uso de redes Wi-Fi abiertas, opción desaconsejada desde el punto de vista de seguridad), siempre que se fuerce a que la conexión de datos de telefonía móvil emplee tecnologías 3/4G frente a 2G (ver apartado "18. Comunicaciones móviles: voz y datos"). Para ello sería necesario deshabilitar el interfaz Wi-Fi y habilitar las comunicaciones móviles de datos.

1605. En caso contrario, es decir, si se requiere cursar todo el tráfico de datos a través de redes Wi-Fi (con un nivel de seguridad adecuado) se recomienda consultar el apartado "18. Comunicaciones móviles: voz y datos", que contiene información detallada de cómo deshabilitar el interfaz de datos de telefonía móvil 2/3/4G.

1606. Cuando ambas conexiones de datos han sido activadas, a través del comando `ifconfig` (vía `adb shell`) es posible ver que sólo el interfaz Wi-Fi (de mayor prioridad), tiene asignada una dirección IP. El interfaz asociado a las capacidades de datos de telefonía móvil (`rmnet_data0`), anteriormente denominado `rmnet0`, se encuentra en estado inactivo (o `DOWN`), motivo por el que no es mostrado en la imagen inferior, al estar activo (o `UP`) el interfaz Wi-Fi (`wlan0`).

```

rmnet_ipa0 Link encap:UNSPEC
  UP RUNNING MTU:2000 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:0 TX bytes:416

wlan0      Link encap:UNSPEC
  inet addr:172.16.16.101 Bcast:172.16.16.255 Mask:255.255.255.0
  inet6 addr: fe80::2e59:8aff:fe6f:f2ec/64 Scope: Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:1330 errors:0 dropped:0 overruns:0 frame:0
  TX packets:389 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:3000
  RX bytes:244979 TX bytes:58229

dummy0     Link encap:UNSPEC
  inet6 addr: fe80::3008:e2ff:fe80:a081/64 Scope: Link
  UP BROADCAST RUNNING NOARP MTU:1500 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:0 TX bytes:210

p2p0       Link encap:UNSPEC
  UP BROADCAST MULTICAST MTU:1500 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:3000
  RX bytes:0 TX bytes:0

r_rmnet_data0 Link encap:UNSPEC
  inet6 addr: fe80::a3c4:1d39:107a:2894/64 Scope: Link
  UP RUNNING MTU:3000 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:0 TX bytes:416

lo         Link encap:UNSPEC
  inet addr:127.0.0.1 Mask:255.0.0.0
  inet6 addr: ::1/128 Scope: Host
  UP LOOPBACK RUNNING MTU:65536 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:0 TX bytes:0

```

Nota: En versiones anteriores de Android (hasta la 5.x inclusive), existía el comando *netcfg*, que permitía visualizar la configuración de las redes de forma rápida. El comando *netcfg* se eliminó en Android 6.x¹⁰¹, y se reemplazó por el comando *ifconfig* (heredado de Linux).

1607. En el caso del interfaz Wi-Fi ("wlan0"), cuando se deshabilita, dejará de mostrarse en la salida del comando *ifconfig* (al no encontrarse en estado activo o "UP"). El interfaz asociado a las capacidades de datos de telefonía móvil ("rmnet_data0") pasará a tener asociada la dirección IP que le otorgue el proveedor del servicio de telefonía por DHCP, y el tráfico entre el dispositivo móvil e Internet se cursará a través suyo:

```
rmnet_data0 Link encap:UNSPEC
```

¹⁰¹ https://github.com/bbc/device_api-android/issues/92

```

inet addr:10.25.250.182  Mask:255.255.255.252
inet6 addr: fe80::737b:8e43:67da:cc73/64 Scope: Link
UP RUNNING MTU:1340 Metric:1
RX packets:22 errors:0 dropped:0 overruns:0 frame:0
TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2274 TX bytes:4488

```

1608. Mediante la tabla de rutas ("/proc/net/route") es posible validar el comportamiento descrito y el interfaz prioritario para el intercambio de datos. La parte superior de la siguiente imagen muestra como el tráfico es dirigido a través del interfaz Wi-Fi ("wlan0") pese a estar ambos interfaces de red activos, empleando el router por defecto para la red 172.16.17.0 (en hexadecimal, y con los valores de los octetos invertidos, 0x001110AC); la parte inferior, muestra cómo el tráfico se enruta a través del interfaz de telefonía móvil ("rmnet_data0") empleando el router por defecto para la red 10.25.250.180 (en hexadecimal, y con los valores de los octetos invertidos, 0xB4FA190A):

```

bullhead:/ $ cat /proc/net/route

```

Iface	Destination	Gateway	Flags	RefCnt	Use	Metric	Mask
MTU	Window	IRTT					
0	0	0					
wlan0	<u>001110AC</u>	00000000		0001	0	0	0
00FFFFFF	0	0	0				

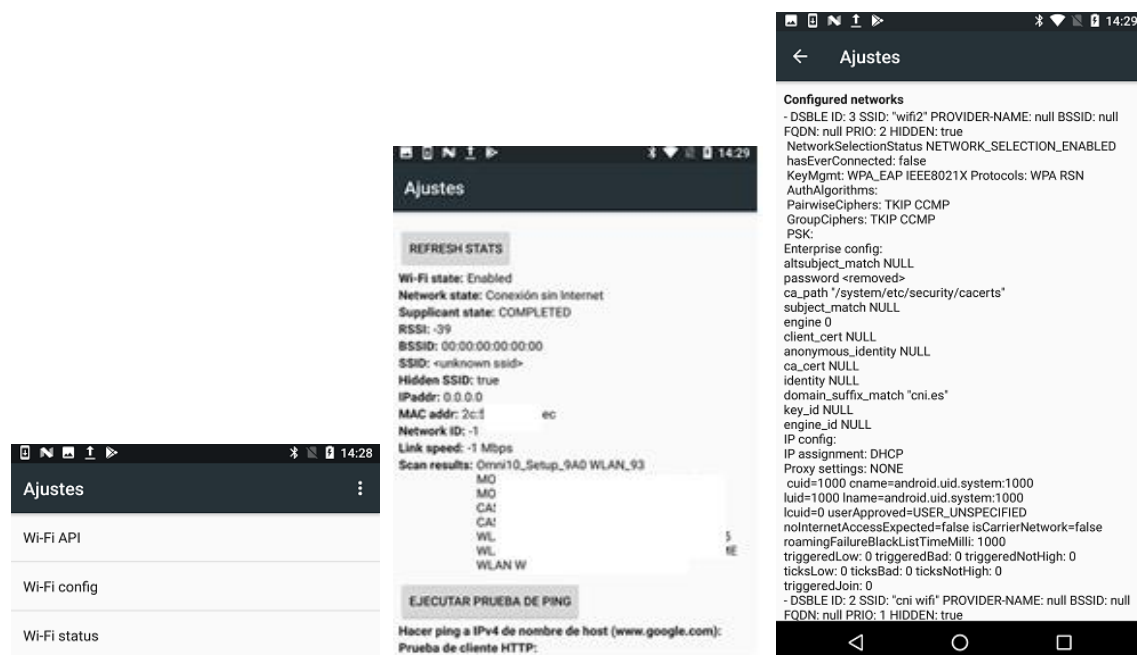
```

bullhead:/ $ cat /proc/net/route

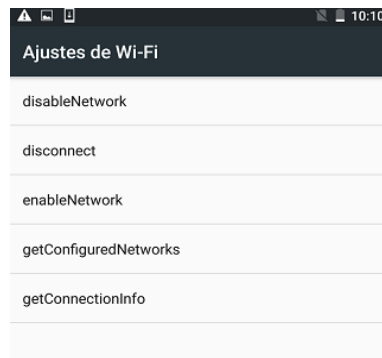
```

Iface	Destination	Gateway	Flags	RefCnt	Use	Metric	Mask
MTU	Window	IRTT					
rmnet_data0	<u>B4FA190A</u>	00000000		0001	0	0	0
FCFFFFFF	0	0	0				

1609. Adicionalmente, en el menú de "Prueba" (o *testing*) accesible mediante la marcación del código *****4636***** en el dispositivo móvil, se ofrece el menú "Wi-Fi information", que a su vez contiene las opciones "Wi-Fi status" (ver imagen inferior central), "Wi-Fi Config" (ver imagen derecha) y "Wi-Fi API":

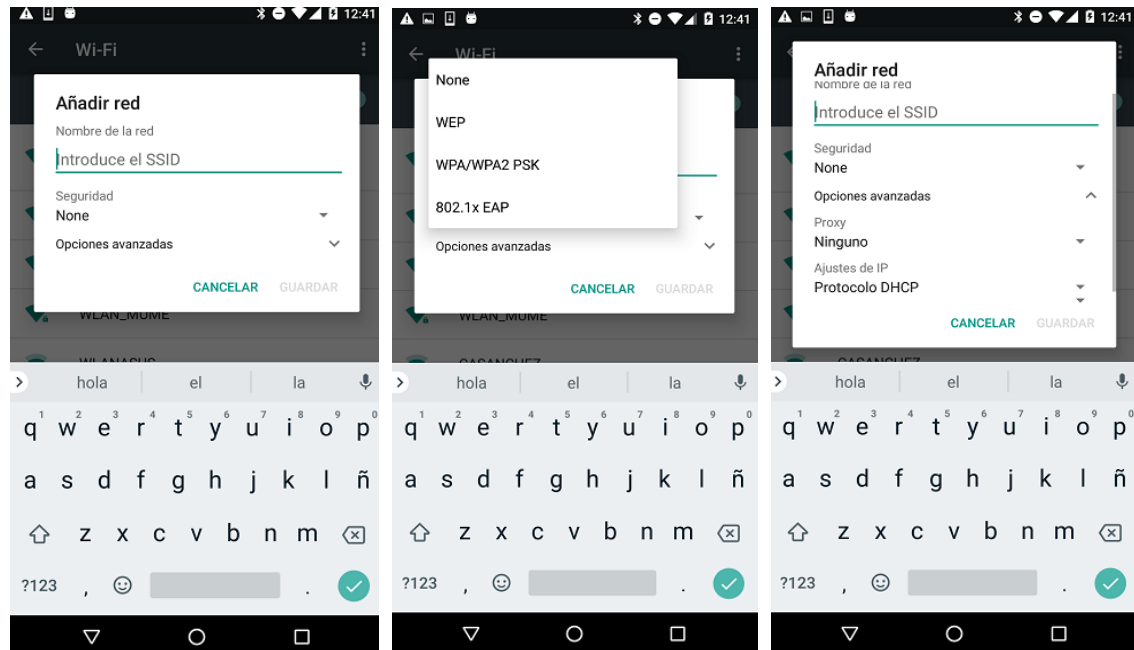


1610. La opción "Wi-Fi API" alberga opciones que actúan como comandos sobre el interfaz Wi-Fi. Es importante reseñar que se trata de opciones no soportadas, por lo que no se recomienda recurrir a ellas. Por ejemplo, "disableNetwork" deshabilita la red cuyo identificador se proporcione; "disconnect" desconecta el interfaz Wi-Fi por completo; "enableNetwork" habilita la red que se pase como parámetro; "getConfiguredNetworks" y "getConnectionInfo" no funcionan con la versión 6.0.1 en el dispositivo Nexus 5X utilizado en la elaboración de la presente guía.

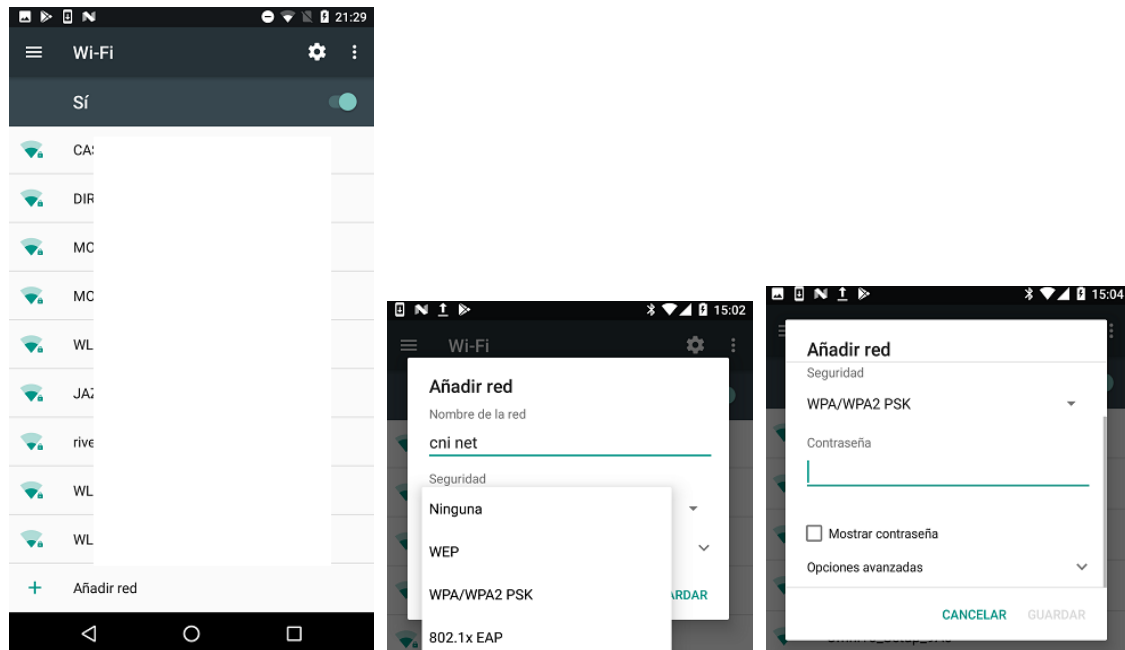


16.3. RECOMENDACIONES DE SEGURIDAD PARA LA CONEXIÓN A REDES WI-FI

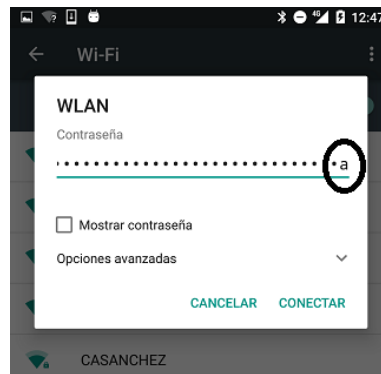
1611. Este apartado proporciona recomendaciones de configuración a través de los diferentes parámetros y opciones disponibles en Android para la conexión segura a redes Wi-Fi.
1612. La configuración manual y detallada de las redes Wi-Fi se puede realizar desde el menú "Ajustes - Wi-Fi", y en concreto mediante la opción de añadir una red Wi-Fi manualmente, a través del menú "Añadir red" de la parte superior (disponible con el interfaz Wi-Fi activo).
1613. Esta opción permite seleccionar todas las opciones de configuración de conexión de una nueva red Wi-Fi, incluyendo el nombre de red (o SSID), y el mecanismo de seguridad a emplear (abierta o ninguna, WEP, WPA/WPA2 PSK o 802.1x EAP; todos ellos analizados posteriormente). La opción "Opciones avanzadas" permite seleccionar el proxy y la configuración de la dirección IP:



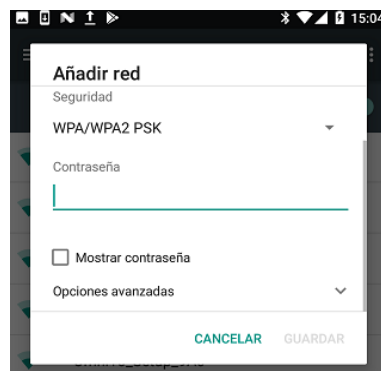
1614. Adicionalmente, mediante la selección de una de las redes disponibles al alcance del dispositivo móvil (listadas en la misma pantalla bajo la sección "Wi-Fi") es posible seleccionar la red Wi-Fi con la que se desea establecer la conexión.
1615. Como se explicará posteriormente en detalle, desde el punto de vista de seguridad se recomienda seleccionar la red Wi-Fi con la que se quiere establecer la conexión de la lista de redes visibles, en lugar de añadirla manualmente a través del menú "Añadir red".
1616. Habitualmente se recomendaba hacer uso del mecanismo de configuración manual frente a seleccionar la red Wi-Fi desde el listado de redes disponibles, ya que, en este segundo caso, no se dispone de tanta flexibilidad y granularidad para seleccionar los diferentes parámetros de configuración y los mecanismos de seguridad de la red. Sin embargo, en el caso de Android se recomienda añadir la red Wi-Fi desde el listado de redes disponibles (ver apartado "16.5. Conexión automática a redes Wi-Fi: ocultas o visibles y restricción de conexión") para evitar una vulnerabilidad existente al añadir la red Wi-Fi de forma manual.
1617. Las redes aparecen clasificadas en la lista (por defecto) en orden según la intensidad de señal de cada una de ellas, siendo las redes de la zona superior las de mayor intensidad de señal (de mayor a menor intensidad). La lista permite de manera muy limitada identificar los mecanismos de seguridad empleados por cada una de la redes Wi-Fi mediante el icono disponible junto al nombre de cada red:



1618. La existencia de algún mecanismo o tipo de seguridad en cada red Wi-Fi viene únicamente reflejado por un icono de un candado, pero no se indica el tipo de seguridad concreto empleado por la red.
1619. Para establecer la conexión con una red próxima únicamente es necesario introducir la contraseña de acceso a la red (ver imagen superior derecha) en caso de estar protegida por una contraseña (WEP o WPA/WPA2 PSK). En el caso de redes abiertas directamente se establece la conexión con la red una vez ha sido seleccionada por parte del usuario de la lista (o una vez se ha proporcionado su nombre si se añadiera manualmente).
1620. Todas las redes que implementan algún tipo de mecanismo de seguridad, WEP, WPA o WPA2, aparecen con un candado en el icono de la red y, a partir de Android 5.x no se puede diferenciar el mecanismo de seguridad empleado específicamente desde el listado de redes disponibles, tal como ocurría en versiones previas de Android, como 4.4.x, donde se proporcionaban los detalles bajo el nombre de la red, y se permitía diferenciar, por ejemplo, entre redes WPA y WPA2.
1621. Al establecer una primera conexión con una nueva red Wi-Fi, se recomienda agregar la nueva red Wi-Fi de forma automática a través del listado de redes disponibles (debido a una vulnerabilidad al añadirla manualmente que se describirá posteriormente), pese a que no sea posible visualizar y especificar todas las opciones de configuración disponibles que se detallan a continuación.
1622. El proceso de configuración de la contraseña puede permitir al usuario no ocultar el texto introducido en pantalla (por ejemplo, con el carácter ".") mediante la opción "Mostrar contraseña" (no habilitada por defecto). Hay que señalar, no obstante, que cada carácter que se introduce se muestra brevemente en pantalla, por lo que conviene prestar atención y evitar configurar la red si no se dispone de la privacidad suficiente.



1623. En caso de habilitar la opción de "Mostrar contraseña", desaconsejada desde el punto de vista de seguridad, el usuario debe prestar atención a cómo y dónde introduce la contraseña de conexión a la red Wi-Fi para evitar que un potencial atacante la obtenga mediante acceso visual a la pantalla del dispositivo móvil.
1624. Dentro de las opciones de configuración de una nueva red Wi-Fi, la primera pantalla de configuración hace posible especificar el nombre de la red (SSID) y el mecanismo de seguridad a emplear: Ninguna (opción por defecto), WEP, WPA/WPA2 PSK o 802.1x EAP.
1625. En caso de seleccionar una red WEP o WPA/WPA2 PSK, es necesario especificar la contraseña de la red:



1626. Desafortunadamente, Android no permite especificar detalladamente los mecanismos de autenticación y de cifrado de datos, no siendo posible diferenciar entre redes WPA y redes WPA2 (tanto Personal, PSK, como Empresarial, o Enterprise o 802.1x), ni el tipo de cifrado: TKIP, AES o ambos.
1627. Por otro lado, tampoco es posible especificar si se permiten conexiones a redes con infraestructura, es decir, puntos de acceso, a redes entre equipos o ad-hoc, o ambas.
1628. Android por defecto sigue sin permitir conexiones Wi-Fi ad-hoc (IBSS, Independent Basic Service Set), permitiendo únicamente conexiones a redes Wi-Fi con infraestructura, es decir, basadas en puntos de acceso. La pantalla que muestra el listado de redes Wi-Fi disponibles directamente no muestra las redes ad-hoc.

Nota: Esta limitación del gestor de conexiones Wi-Fi de Android está implícita en el diseño del sistema y es conocida desde las primeras versiones de Android [Ref.- 39].

Aunque desde el punto de vista de seguridad es preferible conectar a redes Wi-Fi con infraestructura, dado que hoy en día es también posible hacer uso de redes ad-hoc con un nivel de seguridad razonable, existe la posibilidad de modificar este comportamiento existente por defecto y habilitar el soporte para redes *ad-hoc* en dispositivos móviles Android *rooteados* a través de un conjunto de parches y modificaciones disponibles, por ejemplo, en distribuciones de Android de terceros, como LineageOS (previamente CyanogenMod [Ref.- 40]).

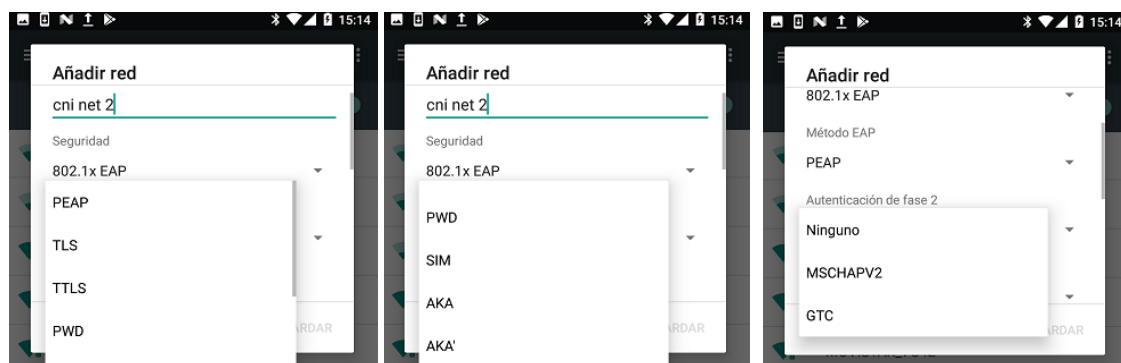
Las versiones 2.3, 4.0 (y posteriores) de Android implementan soporte para Wi-Fi Direct ("P2P", ver apartado "16.7. Wi-Fi Direct ("P2P")"), un estándar de comunicación Wi-Fi directa entre dispositivos que es similar (pero no sustituye completamente) a las comunicaciones *ad-hoc* [Ref.- 41], y añade mejoras de seguridad, facilidad de configuración, rendimiento y consumo.

1629. Igualmente, el proceso de configuración de las redes Wi-Fi en Android no permite claramente, y de forma explícita, especificar o comprobar si la red Wi-Fi es oculta o visible (ver apartado "1650. En caso de que sea necesario conectarse temporalmente a redes inalámbricas no ocultas (ver apartado "16.5. Conexión automática a redes Wi-Fi: ocultas o visibles y restricción de conexión"), o que, siendo ocultas, no sean de uso reiterado, se aconseja proceder a borrar la red a través del botón "Olvidar" para eliminar su almacenamiento y configuración del dispositivo móvil Android, y evitar así ataques basados en la suplantación de redes Wi-Fi cuyo nombre (o SSID) sea conocido por el dispositivo móvil.

1630. Conexión automática a redes Wi-Fi: ocultas o visibles").

1631. Se recomienda conectarse únicamente a redes que emplean WPA2 (Personal, WPA2-PSK, o Enterprise, WPA2) como mecanismo de autenticación, y AES como mecanismo de cifrado. En su defecto, debería emplearse WPA/TKIP (desaconsejado actualmente), pero en ningún caso se debe hacer uso de WEP.

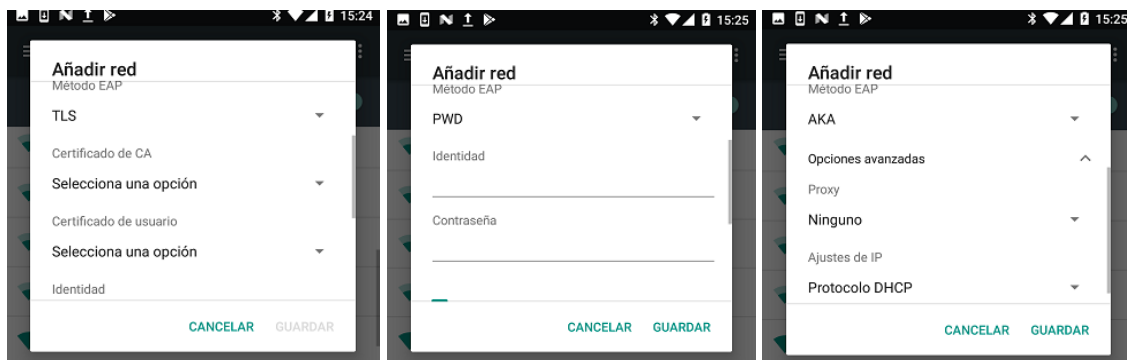
1632. En caso de emplear WPA2 (o WPA) en su versión Enterprise, es decir, con mecanismos basados en 802.1x/EAP, la pantalla de ajustes permite especificar los parámetros de configuración asociados al tipo de mecanismo EAP a emplear: PEAP, TLS, TTLS, PWD¹⁰², SIM, AKA o AKA' y el método de autenticación de fase 2 (Ninguno, MSCHAPV2 (PEAPv0) o GTC, (EAP-GTC, *Generic Token Card*)):



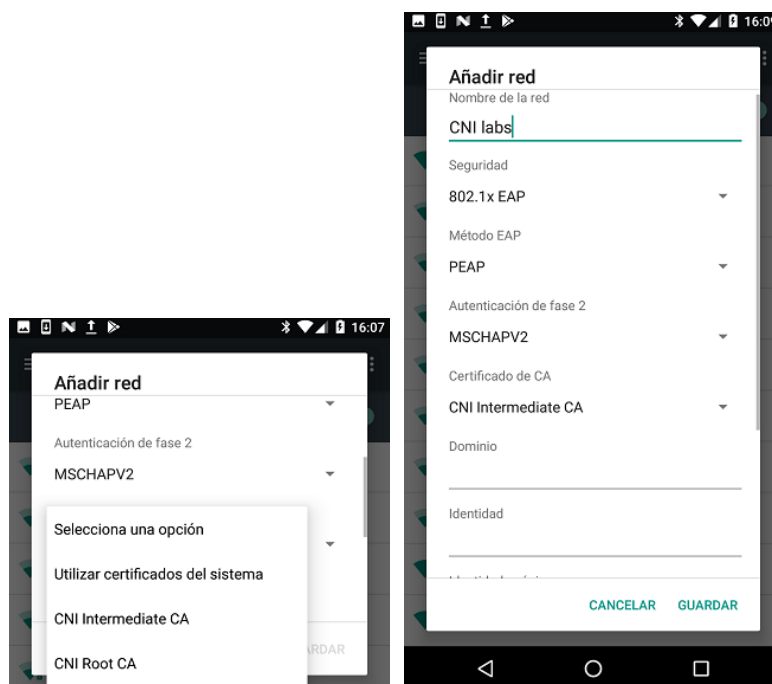
¹⁰² Los tres últimos tipos de métodos EAP corresponden según los estándares a EAP-TLS, EAP-TTLS y EAP-PWD.

Nota: En la versión de Android 5.x únicamente se soportaban los métodos PEAP, TLS, TTLS y PWD para las redes 802.1x EAP (ver imagen superior izquierda). A partir de Android 6.x, se incorporan también los métodos SIM (*Subscriber Identity Module*, empleado para autenticación e intercambio de claves a través de redes GSM), AKA (*Authentication and Key Agreement*, para autenticación e intercambio de claves a través de UMTS) y AKA' (variante del AKA para redes 3G).

1633. En función del tipo de método EAP seleccionado, aparecerán las diferentes opciones según el tipo de autenticación requerido:



1634. Adicionalmente, para PEAP (Protected EAP) debe definirse el certificado digital raíz de la autoridad certificadora (CA), raíz o intermedia, que ha emitido el certificado digital del servidor de autenticación de la red Wi-Fi (RADIUS), mediante el campo "Certificado de CA", que debe haber sido añadido previamente al dispositivo móvil, bien desde el menú de ajustes avanzados de Wi-Fi, bien desde el menú de "Seguridad - Almacenamiento de credenciales" (ver apartado "11.3. Gestión de certificados digitales").



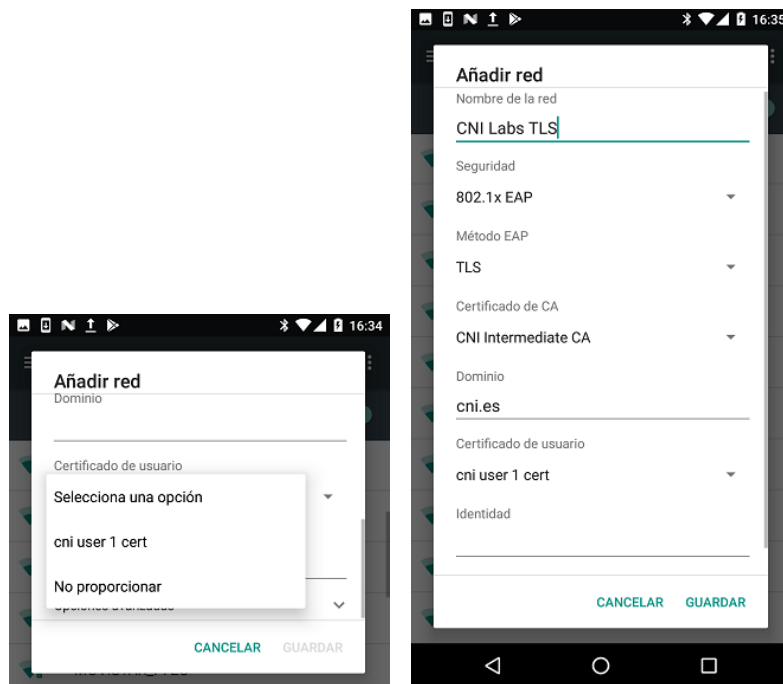
1635. El atributo "Dominio" no existe en el estándar Wi-Fi asociado a 802.1x/EAP; sin embargo, Android 7.x hace que este parámetro sea obligatorio. El valor es un sufijo y debe coincidir con un dominio o subdominio correspondiente a los utilizados por el nombre del servidor

RADIUS que proporciona el certificado digital, es decir, el CN (Common Name) del certificado. Por ejemplo, si el certificado digital pertenece a "radius.cni.es", el valor de "Domino" puede ser "cni.es". El concepto de sufijo del dominio es empleado por la especificación Wi-Fi Hotspot 2.0 de la Wi-Fi Alliance para los certificados de los servidores AAA (ej. RADIUS)¹⁰³.

1636. Por otro lado, para TTLS (conocido como EAP-TTLS) es necesario definir los mismos elementos que para PEAP, pero haciendo uso de la identidad anónima a utilizar para la autenticación externa o de fase 1, mediante el campo "Identidad anónima" (por ejemplo, con el valor estándar "anonymous").
1637. Por último, para TLS (conocido como EAP-TLS) es necesario definir los mismos elementos que para PEAP, pero sustituyendo las credenciales tradicionales del usuario (nombre de usuario y contraseña) por un certificado digital cliente personal y que identifica al usuario, mediante el campo "Certificado de usuario", y que también debe haber sido añadido previamente al dispositivo móvil (ver apartado "11.3. Gestión de certificados digitales").
1638. En versiones anteriores de Android, si se utilizaban certificados emitidos por una de las CAs públicas ya existentes por defecto en el dispositivo móvil, era necesario importarlas manualmente, ya que no eran mostradas por defecto en el campo "Certificado de CA". Sin embargo, Android 7.x ofrece la opción "Utilizar certificados del sistema" para este propósito.
1639. En el caso de emplear EAP-TLS siempre es necesario importar el certificado digital personal del cliente (en formato PKCS#12) en el dispositivo móvil. Para instalar el certificado personal (ver apartado "11.3. Gestión de certificados digitales") debe poderse acceder desde el dispositivo móvil al fichero del certificado digital personal (".pfx" o ".p12") correspondiente.

103

[https://developer.android.com/reference/android/net/wifi/WifiEnterpriseConfig.html#setDomainSuffixMatch\(java.lang.String\)](https://developer.android.com/reference/android/net/wifi/WifiEnterpriseConfig.html#setDomainSuffixMatch(java.lang.String))



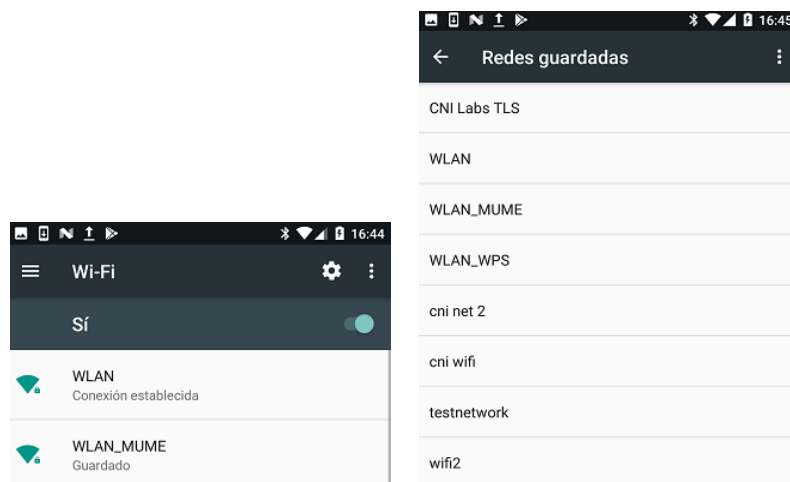
1640. Una vez importado en el dispositivo móvil, el certificado personal estará disponible a través del menú de configuración de la red Wi-Fi 802.1x EAP basada en TLS en el campo "Certificado de usuario" (ver imágenes superiores).

Nota: El certificado digital cliente (o de usuario) es identificado por defecto con un *string* con letras y números, correspondientes al identificador de la clave, si no se modificó su nombre asociado durante el proceso de importación (ver apartado "11.3. Gestión de certificados digitales"). En caso de que se modifique su nombre, el certificado es mostrado con el nombre elegido por el usuario. Android permite importar el mismo certificado digital cliente con diferentes nombres múltiples veces.

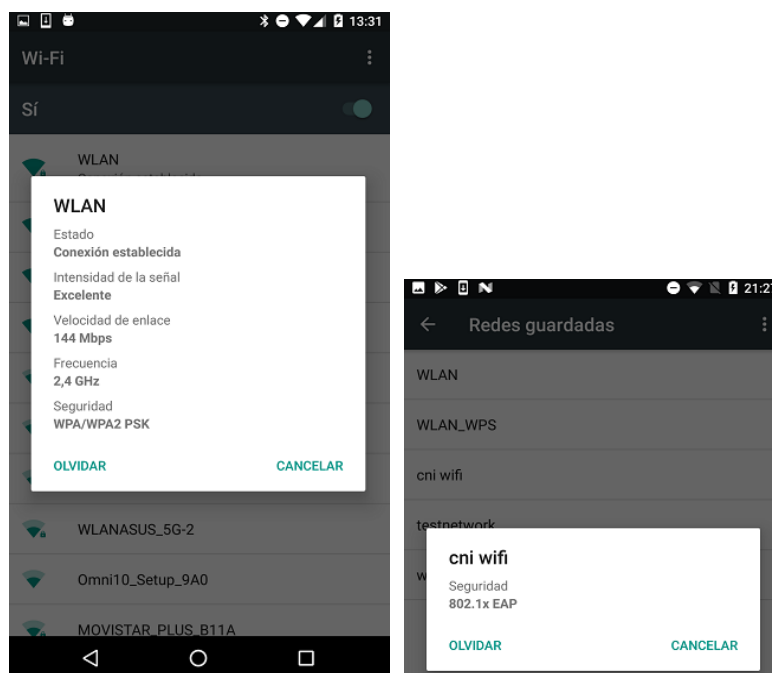
16.4. LISTA DE REDES PREFERIDAS (PNL)

1641. La lista de redes preferidas (PNL, *Preferred Network List*), es decir, a las que se ha conectado previamente el dispositivo móvil o que han sido configuradas en el mismo (ver apartado "16.5. Conexión automática a redes Wi-Fi: ocultas o visibles y restricción de conexión"), no puede ser visualizada tras habilitar el interfaz Wi-Fi, desde el menú "Ajustes - Wi-Fi", y en concreto, al final de la lista de redes disponibles actualmente, mediante el texto "Fuera de rango" (tal como ocurría en versiones previas de Android, como 4.4.x).
1642. En su lugar, las redes almacenadas en la PNL en Android 7.x están disponibles a través de las opciones avanzadas, mediante el menú "Redes guardadas". Las redes Wi-Fi son identificadas por su nombre o SSID.
1643. Las redes existentes en la PNL que están disponibles actualmente (suponiendo que el dispositivo móvil está conectado a otra red conocida, por ejemplo a la red Wi-Fi "WLAN") aparecerán en la lista de redes disponibles con un distintivo que las identifica como redes con las que previamente se ha establecido una conexión, mediante el texto "Guardado"

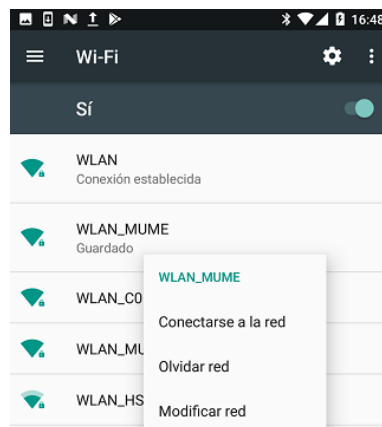
(por ejemplo, la red "WLAN_MUME" en la imagen inferior), salvo que se trate de la red Wi-Fi a la que está actualmente conectado el dispositivo móvil, ya que ésta aparecerá con el texto "Conexión establecida":



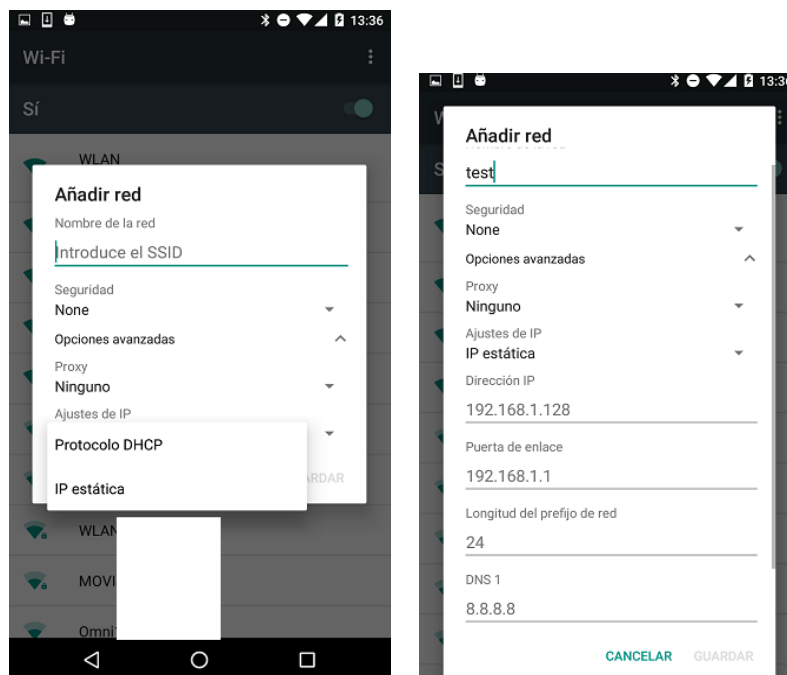
1644. Mediante la selección de una red existente en la PNL desde la pantalla con el listado de redes Wi-Fi (con una pulsación breve sobre la misma), es posible eliminarla de la PNL a través del botón "Olvidar" (tanto si es visible actualmente y hay establecida una conexión con ella, imagen izquierda, como si es visible pero no hay una conexión establecida, imagen central, o como si no es visible actualmente pero está en la PNL, imagen derecha):



1645. Mediante la selección de una red existente en la PNL desde la pantalla con el listado de redes Wi-Fi (con una pulsación prolongada sobre la misma), es también posible eliminarla de la PNL a través de la opción "Olvidar red" o editar sus detalles (como la contraseña de acceso) a través de la opción "Modificar red" (tanto si no es visible actualmente, como si lo es):



1646. Por defecto, la configuración de TCP/IP empleada para la conexión a las redes Wi-Fi hace uso de una dirección IP obtenida dinámicamente mediante el protocolo DHCP (ver apartado "19. Comunicaciones TCP/IP").
1647. Android permite, a través de la opción "Opciones avanzadas", fijar una dirección IP estática (junto a la puerta de enlace, la máscara o prefijo de red, y los servidores DNS primario y secundario):



1648. Para llevar a cabo la configuración de TCP/IP, o modificaciones en la misma, sí es necesario que el interfaz Wi-Fi esté activo para poder seleccionar la red Wi-Fi de la lista de redes visibles o de la PNL.
1649. En resumen, desde el punto de vista de seguridad de Android como cliente Wi-Fi, se recomienda hacer uso de redes inalámbricas no ocultas (ver apartado "16.5. Conexión automática a redes Wi-Fi: ocultas o visibles y restricción de conexión"), basadas en WPA2 con cifrado AES, y autenticación de tipo Personal, con una contraseña de acceso (PSK, *Pre-Shared Key*) suficientemente robusta (de más de 20 caracteres), o autenticación de tipo Enterprise, basada en mecanismos de autenticación 802.1x/EAP, y empleando EAP-TLS (con certificados digitales cliente) preferiblemente, y correctamente configurada y gestionada.

Nota: Con el objetivo de proteger el dispositivo móvil y aumentar su nivel de seguridad, se desaconseja el uso de redes inalámbricas públicas y abiertas, con un nivel de seguridad reducido o inexistente (sin mecanismos de autenticación y cifrado), como por ejemplo *hotspots* Wi-Fi disponibles en hoteles, aeropuertos o centros de conferencias [Ref.- 194].

Los riesgos de seguridad asociados a este tipo de entornos incluyen la captura e interceptación del tráfico y datos por parte de un potencial atacante, la posibilidad de inyección de tráfico para la realización de ataques directos y de accesos no autorizados contra el dispositivo móvil, ataques de suplantación de la red, y ataques de denegación de servicio (DoS), entre otros.

1650. En caso de que sea necesario conectarse temporalmente a redes inalámbricas no ocultas (ver apartado "16.5. Conexión automática a redes Wi-Fi: ocultas o visibles y restricción de conexión"), o que, siendo ocultas, no sean de uso reiterado, se aconseja proceder a borrar la red a través del botón "Olvidar" para eliminar su almacenamiento y configuración del dispositivo móvil Android, y evitar así ataques basados en la suplantación de redes Wi-Fi cuyo nombre (o SSID) sea conocido por el dispositivo móvil.

16.5. CONEXIÓN AUTOMÁTICA A REDES WI-FI: OCULTAS O VISIBLES Y RESTRICCIÓN DE CONEXIÓN

1651. Si el interfaz Wi-Fi está activo y se detecta una red Wi-Fi activa cuyo nombre (o SSID) coincide con el de alguna red existente en la lista de redes preferidas del dispositivo (PNL, *Preferred Network List*), Android intentará conectarse automáticamente dicha red.

1652. La lista de redes preferidas (PNL) está disponible a través del menú "Ajustes - Wi-Fi" (ver apartado "16.4. Lista de redes preferidas (PNL)"). En el caso de redes no disponibles, aparecerán dentro del menú "Redes guardadas". En el caso de redes disponibles, aparecerán a lo largo de la lista de redes (según la intensidad de su señal), identificadas con el texto "Guardado" (si no es la red a la que está conectado el dispositivo móvil actualmente) o "Conexión establecida" (si se trata de la red a la que está conectado el dispositivo móvil actualmente).

1653. La lista de redes preferidas sólo está disponible cuando el interfaz Wi-Fi está habilitado, no siendo posible acceder a ellas (para su borrado o edición) si éste está deshabilitado.

1654. Debe tenerse en cuenta que no existe la opción de desconectarse de la red a la que el dispositivo móvil está conectado actualmente. La única opción disponible es deshabilitar el interfaz Wi-Fi o seleccionar otra red conocida disponible.

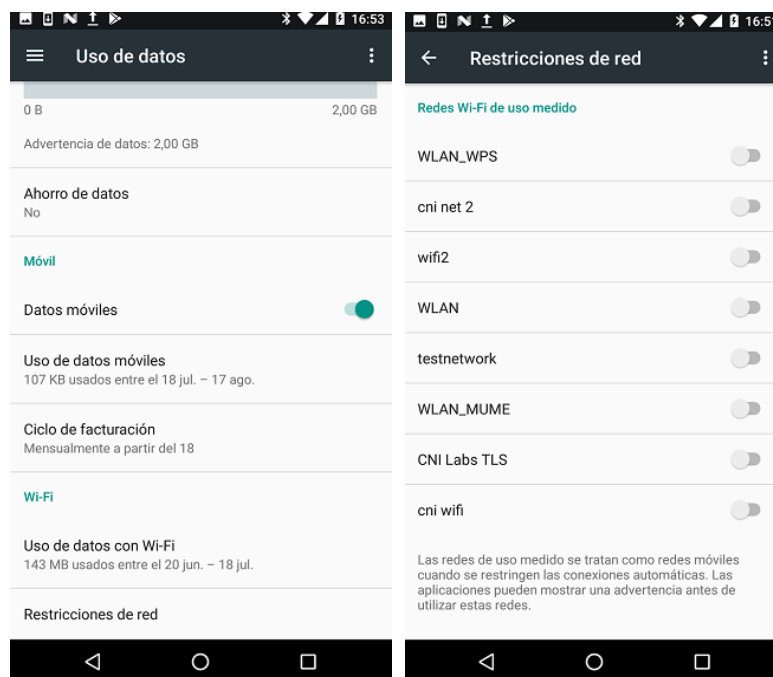
1655. Android no proporciona ninguna opción de configuración global que permita deshabilitar el proceso de conexión automática a redes Wi-Fi si el interfaz Wi-Fi se encuentra activo.

1656. No obstante, Android 6.x introdujo la posibilidad de restringir las conexiones a determinadas redes, de forma que no se genere tráfico hacia ellas ni por parte de apps que ejecuten en segundo plano ni por parte de

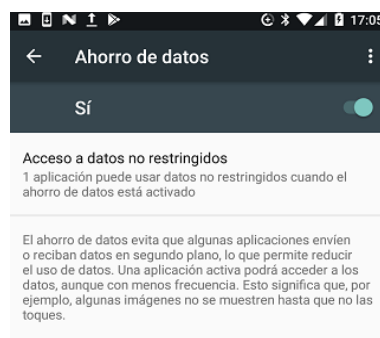
apps que, ejecutando en primer plano, intenten transferir un volumen de datos significativo. Esta opción no implica que no se realicen conexiones a dicha red, pero sí avisará si se intenta descargar actualizaciones de sistema operativo o apps o ficheros de cierto tamaño a través de ella.

1657. La opción para establecer restricciones de red requiere los siguientes pasos:

- Abrir el menú "Ajustes - [Conexiones inalámbricas y redes] Uso de datos - [Wi-Fi] - Restricciones de red":



- Se mostrará una lista de redes guardadas, que son aquellas a las que el dispositivo ha intentado conectarse alguna vez, con o sin éxito. A la derecha de cada red, se dispone de un interruptor, a través del cual se determinará si dicha red se incluirá en la lista de redes restringidas cuando se habilite la opción "Restringir conexiones automáticas" (descrita a continuación).
- Una vez seleccionadas las redes Wi-Fi sobre las que se desea restringir las conexiones, se retrocederá al menú anterior de "Uso de datos" y se seleccionará "Ahorro de datos". Esta opción, que en Android 6.x se configuraba a través de la opción "Uso de datos - [...] - Restringir conexiones automáticas", provoca que las aplicaciones envíen o reciban datos cuando ejecutan en segundo plano:

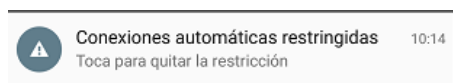


- La opción de restringir conexiones automáticas podría resultar también útil cuando se comparte la conexión de datos del terminal con otro dispositivo. Esta compartición de datos, como se describe en el apartado "16.6. Punto de acceso o zona Wi-Fi", se lleva a cabo a través de una conexión Wi-Fi que se establece entre ambos dispositivos, aunque lo que se comparta en realidad sea la conexión de datos móviles a través de un punto de acceso Wi-Fi creado por el dispositivo móvil.
- Aunque no se indica que sea necesario reiniciar el interfaz Wi-Fi para que los cambios tengan efecto, se han observado escenarios que sí lo precisan, por lo que se recomienda hacerlo para garantizar que los cambios en los ajustes se aplican correctamente.

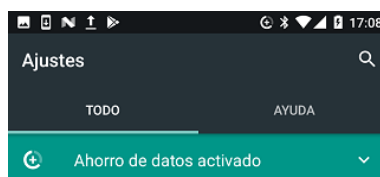
1658. Un ejemplo de funcionamiento de este escenario de conexión restringida a una red de datos o Wi-Fi es el de la aplicación "Fotos", que no realizará su copia de seguridad sobre redes Wi-Fi que se hayan definido como de uso medido (*metered*) si se ha habilitado la opción "Ahorro de datos". El mensaje de notificación que se mostrará es "Esperando red Wi-Fi", pese a que el dispositivo móvil está actualmente a una red Wi-Fi restringida:



1659. En Android 6.x, se avisa de la existencia de la restricción de conexiones automáticas a través de una notificación general que no se puede descartar. Aunque las apps podrían avisar mediante algún mensaje o notificación propio en caso de que alguna de sus actividades haya quedado bloqueada a causa de dicha restricción, no siempre informan apropiadamente de manera particular, por lo que, en caso de observarse comportamientos anómalos, se recomienda revisar esta opción (y la notificación asociada):



1660. En Android 7.x no se muestra esta notificación. Cuando el modo de ahorro de datos está activo, en la barra superior de estado se mostrará un icono con un símbolo "+" dentro de una semicircunferencia, pero no se indica específicamente que haya redes afectadas por las restricciones de datos. Adicionalmente, el menú de "Ajustes" proporcionará un mensaje en la sección superior correspondiente:



1661. Dado que no existe una opción de configuración en el interfaz de usuario de Android para deshabilitar el proceso de conexión automática a

redes Wi-Fi si el interfaz Wi-Fi se encuentra activo, lo más recomendable es deshabilitar el interfaz Wi-Fi cuando no se esté haciendo uso del mismo.

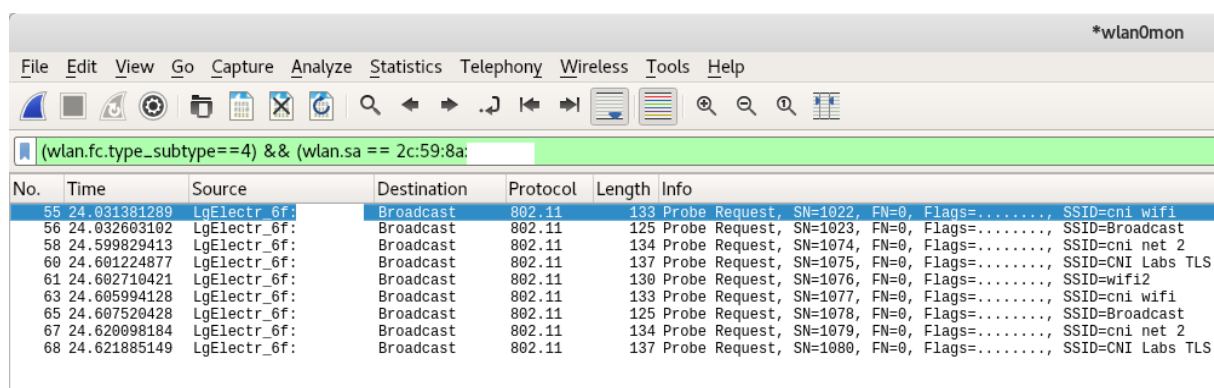
1662. Las redes Wi-Fi pueden ser configuradas en dos modos: oculto o visible. Las redes ocultas son aquellas que no especifican el nombre de la red (SSID, *Service Set Identifier*) en las tramas de anuncio de *broadcast* o *beacons*, redes también conocidas como *nonbroadcast*. Las redes visibles son las que sí incluyen el nombre de red en esas tramas.
1663. Desde el punto de vista de seguridad, y con el objetivo de proteger a los clientes de la red Wi-Fi, se recomienda no hacer uso de redes Wi-Fi ocultas.
1664. Aunque en primera instancia puede parecer que el uso de redes Wi-Fi ocultas es una opción más segura, en realidad no es así, ya que es sencillo para un potencial atacante descubrir su existencia e incluso el nombre de esas supuestas redes ocultas.
1665. Sin embargo, el uso de una red oculta implica reducir el nivel de seguridad en los clientes Wi-Fi que se conectan a ella, como por ejemplo los dispositivos móviles basados en Android. Si una red está oculta, los clientes Wi-Fi se verán obligados a comprobar de forma explícita si está presente, generando tramas (de tipo *probe request*) dirigidas que desvelan necesariamente el nombre de la red, y lo que es más crítico, que les exponen frente a ataques de suplantación de dicha red por parte de un potencial atacante (especialmente cuando la red no emplea mecanismos de seguridad).
1666. Desafortunadamente, Android no implementa ningún mecanismo explícito para definir si una red es oculta o no, por lo que en función de cómo se añada la red Wi-Fi al dispositivo móvil la primera vez, ésta será considerada como una red Wi-Fi oculta o visible.
1667. Si la red Wi-Fi es añadida al dispositivo, y en consecuencia a la PNL, seleccionándola de la lista de redes disponibles (situación que sólo se puede presentar si la red es visible), será añadida como red visible (escenario que aporta mayor seguridad a la configuración de Android).
1668. Si la red Wi-Fi es añadida al dispositivo, y en consecuencia a la PNL, manualmente mediante el menú "Añadir red" disponible en la parte superior (situación necesaria a la hora de añadir redes ocultas), será añadida como red oculta (escenario vulnerable desde el punto de vista de Android).
1669. Este es el comportamiento más vulnerable (existente previamente pero corregido en algunos sistemas operativos y ordenadores portátiles y de sobremesa desde hace años), ya que obliga a Android a generar tráfico que desvela la lista de redes Wi-Fi configuradas en el dispositivo móvil (conocida como la PNL). Este comportamiento presenta un escenario similar al ya existente en equipos de escritorio basados en Windows XP sin la actualización de seguridad KB917021 aplicada, y conocida desde el año 2007 [Ref.- 49].
1670. Adicionalmente, debe tenerse en cuenta que una vez una red Wi-Fi ha sido añadida al dispositivo móvil y está en la PNL, no se dispone de un

mecanismo a través del interfaz gráfico de Android para determinar si es realmente considerada oculta o visible.

1671. Resumiendo: para todas aquellas redes configuradas o añadidas a través del menú "Añadir red", Android generará tráfico que desvelará estas redes Wi-Fi disponibles en su lista de redes preferidas e intentará conectarse a ellas, escenario que puede ser empleado por un potencial atacante para suplantar dicha red, disponer de conectividad con el dispositivo móvil y proceder a explotar vulnerabilidades en el mismo.

1672. Por tanto, no se recomienda añadir ninguna red Wi-Fi manualmente al dispositivo móvil a través del menú "Añadir red", aunque es una práctica común desde el punto de vista de seguridad para disponer de mayor control sobre la configuración y ajustes de la red Wi-Fi a añadir.

1673. En la imagen inferior, el dispositivo móvil Android intenta descubrir la presencia de las redes Wi-Fi disponibles en su PNL, "CNI LABS TLS", "cni net 2", "wifi2" y "cni wifi", una vez que el interfaz Wi-Fi ha sido activado, así como periódicamente, incluso aunque esté conectado a otra red Wi-Fi diferente en la actualidad:



The screenshot shows a Wireshark capture on the wlan0mon interface. The filter is set to (wlan.fc.type_subtype==4) && (wlan.sa == 2c:59:8a:). The packet list shows several broadcast probe requests from the source LgElectr_6f: to various destinations. The packet details for the selected packet (No. 55) show a Probe Request with SN=1022, FN=0, and Flags=....., with SSID=cni wifi.

No.	Time	Source	Destination	Protocol	Length	Info
55	24.031381289	LgElectr_6f:	Broadcast	802.11	133	Probe Request, SN=1022, FN=0, Flags=....., SSID=cni wifi
56	24.032603102	LgElectr_6f:	Broadcast	802.11	125	Probe Request, SN=1023, FN=0, Flags=....., SSID=Broadcast
58	24.599829413	LgElectr_6f:	Broadcast	802.11	134	Probe Request, SN=1074, FN=0, Flags=....., SSID=cni net 2
60	24.601224877	LgElectr_6f:	Broadcast	802.11	137	Probe Request, SN=1075, FN=0, Flags=....., SSID=CNI Labs TLS
61	24.602710421	LgElectr_6f:	Broadcast	802.11	130	Probe Request, SN=1076, FN=0, Flags=....., SSID=wifi2
63	24.605994128	LgElectr_6f:	Broadcast	802.11	133	Probe Request, SN=1077, FN=0, Flags=....., SSID=cni wifi
65	24.607520428	LgElectr_6f:	Broadcast	802.11	125	Probe Request, SN=1078, FN=0, Flags=....., SSID=Broadcast
67	24.620098184	LgElectr_6f:	Broadcast	802.11	134	Probe Request, SN=1079, FN=0, Flags=....., SSID=cni net 2
68	24.621885149	LgElectr_6f:	Broadcast	802.11	137	Probe Request, SN=1080, FN=0, Flags=....., SSID=CNI Labs TLS

1674. Esta vulnerabilidad de revelación de información en función de cómo han sido añadidas las redes Wi-Fi a Android (manualmente o automáticamente desde la lista de redes Wi-Fi) fue confirmada por el Android Security Team (Google), aunque parece que no publicará una actualización para solucionarla [Ref.- 95].

Nota: La versión 2.2 de Android introdujo este cambio en el comportamiento de la pila de comunicaciones Wi-Fi, ya que en versiones previas (versiones previas a 2.2 y a la compilación FRF91 empleada para la elaboración de una versión previa de la presente guía) no era posible conectarse con redes Wi-Fi ocultas (tal como identifica el *bug* 1041 de Android [Ref.- 42]); en el pasado incluso aparecieron apps en Google Play para añadir esta funcionalidad.

En cualquier caso, siempre antes de poder establecer una conexión con una red Wi-Fi oculta es necesario añadirla al dispositivo móvil de forma manual, para que éste conozca su existencia, y pueda preguntar sobre su disponibilidad (empleando el comportamiento vulnerable descrito previamente).

1675. Desafortunadamente, los dispositivos móviles Android no sólo desvelan el contenido de su PNL cuando disponen de redes Wi-Fi ocultas configuradas, sino también en numerosos otros escenarios.

1676. En concreto, el hecho de que el dispositivo móvil empleado desvele su PNL depende de múltiples factores asociados a la versión de Android, modelo de dispositivo móvil, tipo de interfaz Wi-Fi y *chipset* empleado, así como del estado del mismo. Algunos dispositivos móviles desvelan su PNL continuamente, incluyendo tanto las redes Wi-Fi ocultas como visibles existentes en la misma, mientras que otros lo hacen esporádicamente en función de estos diferentes factores. Aunque los dispositivos móviles más modernos, junto a la utilización de las últimas versiones de Android, deberían ser menos propensos a desvelar de manera frecuente y habitual su PNL, los Nexus 5X con Android 7.1.2 no presentan este comportamiento.

16.6. PUNTO DE ACCESO O ZONA WI-FI

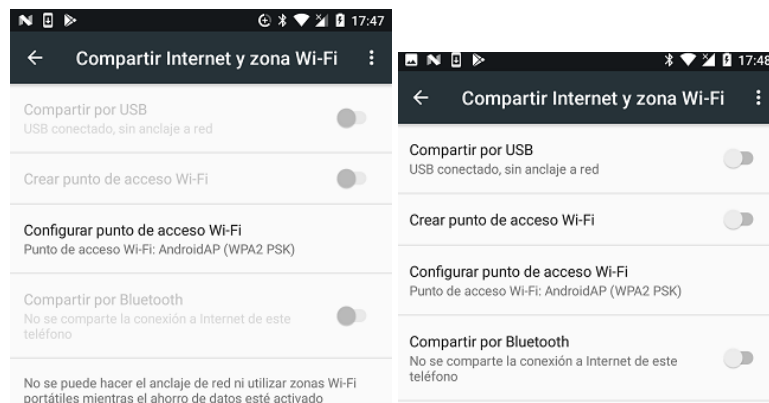
1677. Android dispone de capacidades nativas para actuar como punto de acceso o zona Wi-Fi y router (o enrutador) hacia Internet, proporcionando acceso a múltiples dispositivos (u ordenadores, hasta un máximo de 10) vía Wi-Fi a su conexión de datos de telefonía móvil (2/3/4G) [Ref.- 94]. Para ello es necesario habilitar la conexión de datos móviles previamente (ver apartado "18. Comunicaciones móviles: voz y datos").

1678. Para hacer uso de estas capacidades es necesario acceder al menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Compartir Internet y zona Wi-Fi":

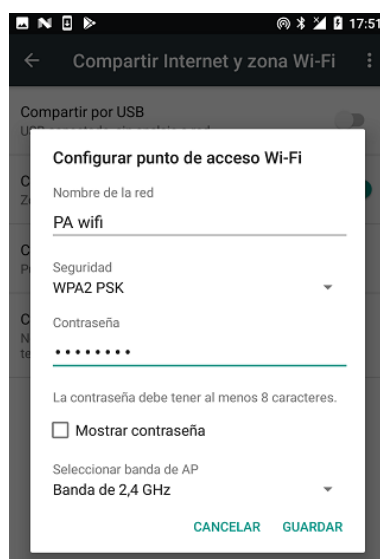


1679. A través de la opción "Compartir punto de acceso Wi-Fi" es posible habilitar el servicio de punto de acceso Wi-Fi (referido en ocasiones como *hotspot* o zona Wi-Fi), de forma que otros ordenadores y dispositivos puedan utilizar el dispositivo móvil como punto de acceso Wi-Fi para conectarse a Internet a través de su conexión de datos de telefonía móvil (2/3/4G).

1680. Si el ahorro de datos está activo, no se permite compartir zonas Wi-Fi con otros dispositivos; aunque a priori parece que sí se permite configurarlo, no se podrá salvar la configuración realizada:



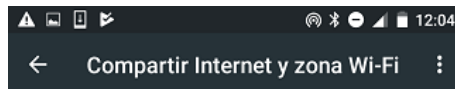
1681. Curiosamente, la zona Wi-Fi puede ser activada en Android incluso aunque el dispositivo móvil no disponga de conexión de datos a través de las redes de telefonía móvil, o incluso sin insertar una tarjeta SIM.
1682. Debe tenerse en cuenta que, a diferencia del interfaz Wi-Fi (cliente) y Bluetooth, para poder llevar a cabo la configuración del punto de acceso Wi-Fi a través del interfaz de usuario del dispositivo móvil no es obligatorio activarlo, por lo que se recomienda modificar y completar su configuración antes de activar la funcionalidad, para evitar potenciales ataques cuando los mecanismos de seguridad no han sido aún aplicados.
1683. Mediante la opción "Configurar zona Wi-Fi", es posible configurar los diferentes ajustes del punto de acceso, incluyendo el nombre de la red Wi-Fi (o SSID) y las características de seguridad:



1684. Por defecto, el punto de acceso Wi-Fi hace uso de WPA2-PSK (*Pre-Shared Key*, o WPA2-Personal), basado en el uso de una contraseña o clave precompartida, opción recomendada desde el punto de vista de seguridad. Android permite también la creación de puntos de acceso Wi-Fi abiertos ("Ninguna"), es decir, que no implementen ningún mecanismo de seguridad (opción desaconsejada).
1685. Por defecto el dispositivo móvil selecciona una contraseña aleatoria de 12 caracteres de longitud, hexadecimales. Sin embargo, se recomienda hacer uso de una contraseña propia, de mayor longitud y que pueda ser

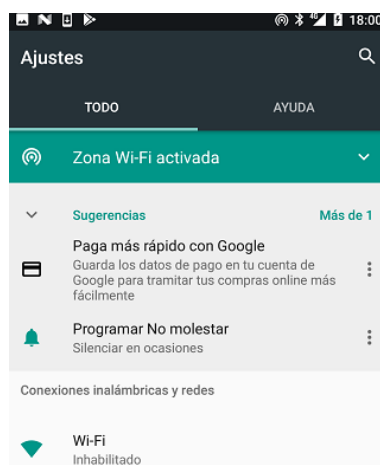
recordada más fácilmente por el usuario. El tamaño mínimo permitido es 8 caracteres.

1686. La habilitación de la zona Wi-Fi se refleja en la barra superior de estado a través del símbolo que se muestra a la izquierda del símbolo de Bluetooth en la imagen inferior:



1687. Cuando el dispositivo móvil actúa de punto de acceso Wi-Fi, las apps de Android no pueden utilizar la conexión Wi-Fi como dispositivo cliente. Si previamente a su activación la conexión Wi-Fi cliente estaba activa, ésta será deshabilitada automáticamente. Igualmente, tras desactivar el punto de acceso Wi-Fi, la conexión Wi-Fi cliente será habilitada de nuevo automáticamente.

1688. En Android 7.x, este comportamiento se constata en la pantalla principal de ajustes, que mostrará el aviso correspondiente en la zona superior, y el estado "Inhabilitado" para el interfaz Wi-Fi:



1689. Debido a la inhabilitación de la conexión Wi-Fi cliente del dispositivo móvil, las aplicaciones que tengan impedido el uso de datos móviles dejarán de poder transferir datos según las restricciones que se hayan definido para este escenario (consultar apartado "18. Comunicaciones móviles: voz y datos").
1690. Se recomienda no habilitar las capacidades de punto de acceso Wi-Fi en el dispositivo móvil, salvo en los casos en que se requiera. En todo caso, este uso debe estar limitado y controlado. En caso de ser utilizada, la red Wi-Fi asociada debe ser protegida y configurada mediante WPA2-PSK con una contraseña o clave precompartida suficientemente larga (más de 20 caracteres) y difícilmente adivinable [Ref.- 48].
1691. El mecanismo de cifrado para WPA2-PSK empleado por defecto por el punto de acceso Wi-Fi proporcionado por Android es AES-CCMP, opción recomendada desde el punto de vista de seguridad.
1692. En caso de activar el punto de acceso Wi-Fi, se recomienda también modificar el nombre de la red Wi-Fi (o SSID) anunciada por el dispositivo

móvil por defecto ("AndroidAP"), con el objetivo de no desvelar que se trata de un punto de acceso basado en Android.

1693. La red Wi-Fi creada es de tipo punto de acceso (o infraestructura), se anuncia en el canal 6 (de 802.11b/g/n), y proporciona incluso capacidades de servidor DHCP para la asignación de direcciones IP a los dispositivos cliente y ordenadores que se conectan a ella.

1694. Tras establecerse la conexión de red a través de la red Wi-Fi creada por Android, y tomando Windows 10 como sistema operativo del ordenador de referencia, el interfaz de red Wi-Fi funciona al estar configurado con una dirección IP dinámica (DHCP), asignándose por defecto una dirección IP de clase C (/24), como por ejemplo 192.168.43.98:

Adaptador de LAN inalámbrica Wi-Fi:

```
Sufijo DNS específico para la conexión. . . :
Dirección IPv4. . . . . : 192.168.43.98
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.43.1
```

```
C:\>ping 192.168.43.1
```

```
Haciendo ping a 192.168.43.1 con 32 bytes de datos:
Respuesta desde 192.168.43.1: bytes=32 tiempo=6ms TTL=64
...
```

1695. El dispositivo móvil (por ejemplo, con dirección IP 192.168.43.1) actúa de gateway, router o pasarela por defecto, de servidor DHCP, y de servidor DNS, y es alcanzable mediante ping (ICMP).

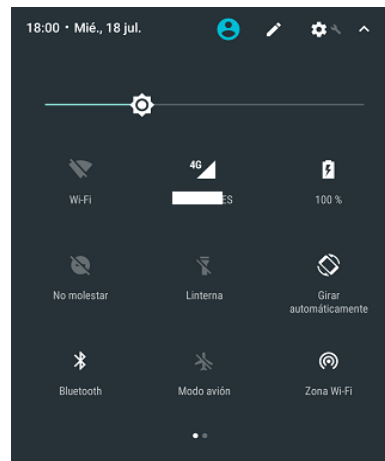
Nota: El direccionamiento IP empleado por Android para el punto de acceso Wi-Fi (192.168.43.x/24) no puede ser modificado salvo en dispositivos móviles *rooteados* [Ref.- 99].

1696. La dirección MAC asociada al interfaz de red del punto de acceso Wi-Fi tiene asociado un OUI con valor "02:1A:11" (por ejemplo), que corresponde a una dirección MAC administrada localmente y, por tanto, no asignada a ningún vendedor públicamente.

1697. Un análisis de los puertos TCP/IP y servicios disponibles en el interfaz de red empleado para el punto de acceso Wi-Fi del dispositivo móvil permite obtener las siguientes conclusiones:

- El único puerto TCP abierto es el correspondiente al servidor DNS (53/tcp). El servidor DNS es identificado por Nmap como "dnsmasq 2.51".
- Los únicos puertos UDP abiertos son los correspondientes al servidor DNS (53/udp) y al servidor DHCP (67/udp).

1698. Es posible interrumpir la disponibilidad del punto de acceso Wi-Fi deshabilitando la opción "Zona Wi-Fi portátil" habilitada previamente (para ello se puede recurrir al menú de ajustes rápidos, como ilustra la imagen inferior), opción recomendada desde el punto de vista de seguridad una vez no se desea disponer de la conexión de red Wi-Fi. Este proceso provocará que el interfaz Wi-Fi se habilite automáticamente de nuevo para uso exclusivo del dispositivo móvil.

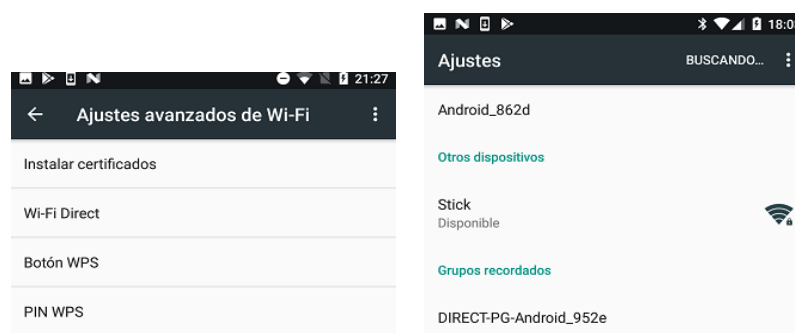


16.7. WI-FI DIRECT ("P2P")

1699. La tecnología Wi-Fi Direct, un estándar publicado por la Wi-Fi Alliance [Ref.- 52], permite el establecimiento de conexiones Wi-Fi directamente entre dispositivos, punto a punto ("P2P", Peer-to-Peer), sin hacer uso de una infraestructura, red o punto de acceso Wi-Fi.
1700. Wi-Fi Direct (y sus servicios asociados) permite a un dispositivo móvil interactuar con otros dispositivos directamente, como por ejemplo mostrar su pantalla y cualquier contenido multimedia (o juegos) en un televisor, monitor o proyector¹⁰⁴, imprimir un documento en una impresora inalámbrica, compartir e intercambiar ficheros con otros dispositivos, como por ejemplo cámaras, o sincronizar datos con un ordenador.
1701. Para hacer uso de Wi-Fi Direct no es necesario disponer de conexión a Internet, aunque esta tecnología también puede ser empleada para compartir el acceso a Internet de un dispositivo con otros dispositivos compatibles con Wi-Fi Direct.
1702. Wi-Fi Direct permite tanto conexiones individuales entre dos dispositivos, como conexiones en grupo de varios dispositivos. Un dispositivo se encargará de proporcionar la red o el grupo Wi-Fi Direct, actuando como punto de acceso (denominado GO, Group Owner) y de gestionar dicha red y su conectividad (por ejemplo, proporcionando un servidor DHCP).
1703. El establecimiento de las conexiones Wi-Fi Direct se puede llevar a cabo mediante la pulsación de un botón, acercando dos dispositivos con capacidades NFC, o mediante la introducción de un PIN, empleando el estándar Wi-Fi Protected Setup (WPS).
1704. La especificación técnica asociada a Wi-Fi Direct, y denominada Wi-Fi Peer-to-Peer ("P2P") define mecanismos para que los dispositivos puedan anunciar y descubrir los servicios ofrecidos por otros dispositivos cercanos a través de Wi-Fi Direct.

¹⁰⁴ Por ejemplo, las conexiones multimedia Miracast pueden ser llevadas a cabo sobre Wi-Fi Direct.

1705. Adicionalmente, a finales de 2014 se ratificó la estandarización de algunas mejoras de los servicios Wi-Fi Direct para llevar a cabo las tareas más comunes de forma sencilla, bajo el principio "descubrir, conectar y hacer" en un sólo paso, como por ejemplo¹⁰⁵: (Wi-Fi Direct) "Send", para el envío de contenidos, "Print", para la impresión de documentos, "for DLNA", para el descubrimiento de dispositivos, y "Miracast", para la transmisión de contenidos multimedia.
1706. La comunicación Wi-Fi Direct hace uso de WPA2 con AES para cifrar y autenticar los datos intercambiados.
1707. Las capacidades de Wi-Fi Direct, en concreto para iniciar una comunicación desde Android, están disponibles a través del menú "Ajustes - Wi-Fi - [...] - Ajustes Avanzados - Wi-Fi Direct", siendo necesario que el interfaz Wi-Fi esté activo:

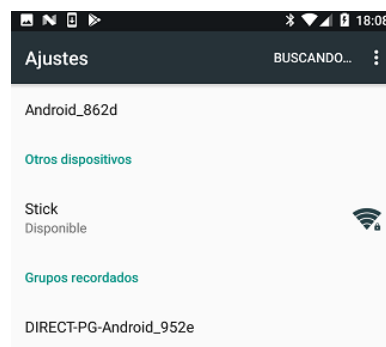


1708. Tras acceder a dicho menú, el dispositivo móvil Android comienza a buscar a otros dispositivos con capacidades Wi-Fi Direct, tal como indica el botón superior ("Buscando...") que se muestra en la imagen superior derecha.
1709. Para ello, envía tramas de tipo *probe request* con el nombre de red, o SSID, "DIRECT-" por defecto, y que desvelan el uso de Wi-Fi Direct. Como origen de estas tramas se emplea una variante de la dirección MAC Wi-Fi del dispositivo móvil, que, por ejemplo, comienza por "7A" en lugar de por "78" (convirtiéndola en una dirección administrada localmente). Por ejemplo, la dirección Wi-Fi que comienza por "78:f8:82" es convertida en "7a:f8:82", respetando el resto de bytes de la dirección.
1710. Adicionalmente, estas tramas 802.11 incluyen un campo asociado a Wi-Fi Direct (o P2P) con el valor "Wi-FiAll", y que contiene detalles de sus capacidades P2P ("P2P Capability"), tanto a nivel de dispositivo como de grupo, y el canal en el que escucha (normalmente, los canales disponibles son 1, 6 y 11 en la banda de frecuencias de 2.4GHz, heredados de los canales disponibles y recomendados en USA):

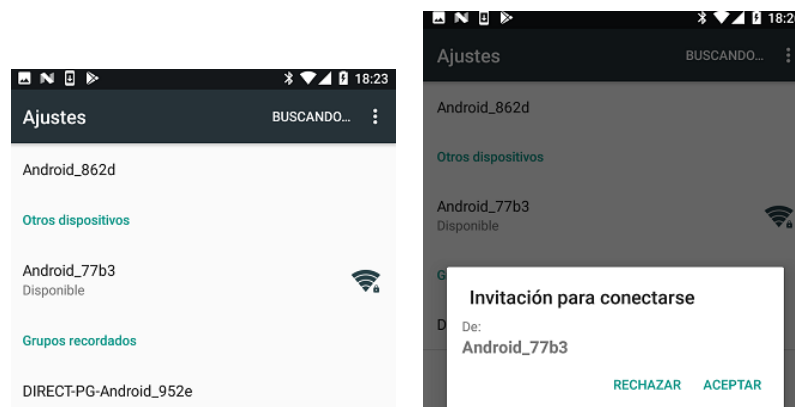
¹⁰⁵ <http://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-enhances-wi-fi-certified-wi-fi-direct>

((wlan.fc.type_subtype==wlan_radio) && (wlan.sa ==						
No.	Time	Source	Destination	Protocol	Length	Info
7	0.112654094	0e	Broadcast	802.11	98	Probe Request, SN=1136, FN=0, Flags=....., SSID=WL...
84	6.059813315		Broadcast	802.11	144	Probe Request, SN=3342, FN=0, Flags=....., SSID=WL
97	8.337987323		Broadcast	802.11	130	Probe Request, SN=643, FN=0, Flags=....., SSID=vod
99	8.341499619		Broadcast	802.11	130	Probe Request, SN=646, FN=0, Flags=....., SSID=vod
1...	10.696786242		Broadcast	802.11	144	Probe Request, SN=335, FN=0, Flags=....., SSID=MOV
1...	12.860574766	0e	Broadcast	802.11	98	Probe Request, SN=1240, FN=0, Flags=....., SSID=WL
1...	12.861924260	0e	Broadcast	802.11	98	Probe Request, SN=1241, FN=0, Flags=....., SSID=WL
1...	12.863233071	0e	Broadcast	802.11	98	Probe Request, SN=1242, FN=0, Flags=....., SSID=WL
1...	12.864764040	0e	Broadcast	802.11	98	Probe Request, SN=1243, FN=0, Flags=....., SSID=WL
2...	19.262934888	0e	Broadcast	802.11	98	Probe Request, SN=1292, FN=0, Flags=....., SSID=WL
3...	25.627333741	0e	Broadcast	802.11	98	Probe Request, SN=1344, FN=0, Flags=....., SSID=WL
3...	25.629580854	0e	Broadcast	802.11	98	Probe Request, SN=1346, FN=0, Flags=....., SSID=WL
3...	25.639709841	0e	Broadcast	802.11	98	Probe Request, SN=1347, FN=0, Flags=....., SSID=WL
3...	28.531797702		Broadcast	802.11	257	Probe Request, SN=1441, FN=0, Flags=....., SSID=DIRECT-
3...	28.545189464		Broadcast	802.11	257	Probe Request, SN=1442, FN=0, Flags=....., SSID=DIRECT-
3...	28.558308102		Broadcast	802.11	257	Probe Request, SN=1443, FN=0, Flags=....., SSID=DIRECT-
3...	30.109946057	HuaweiTe_	Broadcast	802.11	263	Probe Request, SN=756, FN=0, Flags=....., SSID=MOV
3...	32.000684499	HewlettP_	Broadcast	802.11	98	Probe Request, SN=1399, FN=0, Flags=....., SSID=WL

1711. Por defecto, el nombre empleado por Android en Wi-Fi Direct es "Android_<código>", donde "<código>" corresponde con el comienzo del número asociado al nombre o *hostname* empleado por el dispositivo móvil, por ejemplo "android-862d" (ver apartado "7.12. Configuración por defecto del dispositivo móvil"¹⁰⁶), por lo que se recomienda modificarlo a través del botón superior "Cambiar Nombre del Dispositivo" (para Wi-Fi Direct) por un valor que no desvele ningún detalle del terminal, como por ejemplo "dm0001" ("dispositivo móvil 0001"):

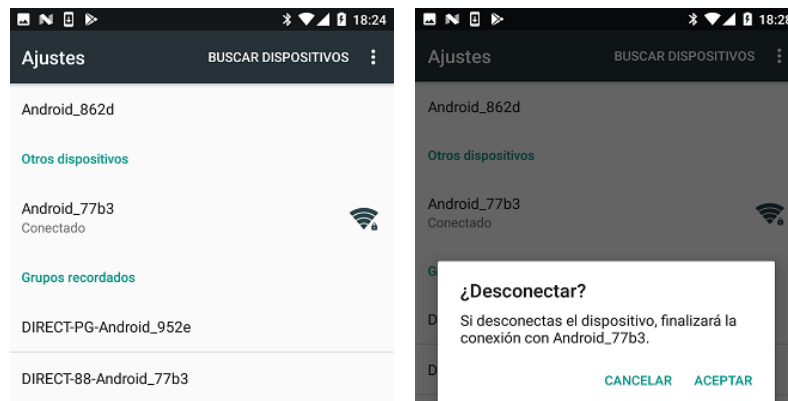


1712. Una vez el dispositivo móvil descubre otros dispositivos con capacidades Wi-Fi Direct, puede iniciar una comunicación enviando una invitación para establecer la conexión. Si el otro dispositivo acepta la invitación, se establecerá un nuevo grupo de Wi-Fi Direct, y se llevará a cabo la conexión:



¹⁰⁶ El valor numérico del nombre cambia aleatoriamente tras restablecer los valores de fábrica.

1713. Asimismo, el dispositivo móvil puede recibir una invitación de comunicación mediante Wi-Fi Direct proveniente de otros dispositivos. El tiempo máximo para aceptar la invitación es de 30 segundos.
1714. Para efectuar la desconexión, basta con pulsar sobre el campo correspondiente al dispositivo y aceptar la confirmación:



1715. Cuando un dispositivo móvil quiere establecer una conexión P2P vía Wi-Fi Direct y envía una invitación, puede, bien solicitar la creación de un grupo al que unirse, bien iniciar la creación de un grupo a través de una trama de "GO Negotiation request" (GO - group owner).
1716. En Android 7.x, cuando se establece la conexión con otro dispositivo vía Wi-Fi Direct, el grupo creado para esa conexión se almacena en el campo "Grupos recordados", de forma que las conexiones futuras desde ese dispositivo se realizarán de forma automática, no recibándose ningún mensaje de confirmación de invitación.
1717. Por tanto, si la conexión Wi-Fi Direct se realiza para una operación puntual, se recomienda eliminar el grupo cuando finalice dicha conexión.
1718. Para el dispositivo móvil empleado en la elaboración de la presente guía, Google no ofrece la opción de crear grupos Wi-Fi Direct, sino que crea un grupo nuevo para cada conexión Wi-Fi Direct aceptada (por tanto, los grupos serán entre dos dispositivos).

17. COMUNICACIONES MÓVILES: MENSAJES DE TEXTO (SMS)

1719. Las capacidades de mensajería de texto (SMS, *Short Message Service*) de los dispositivos móviles permite el envío de mensajes cortos empleando la infraestructura GSM (*Global System for Mobile communications*, o 2G), UMTS (*Universal Mobile Telecommunications System*, o 3G) y LTE (*Long Term Evolution* o 4G).
1720. El primer nivel de protección asociado al sistema de mensajería de texto pasa por la concienciación del usuario desde el punto de vista de seguridad, teniendo en cuenta que la implementación de este módulo en los dispositivos móviles actuales proporciona numerosas funcionalidades, y no únicamente la visualización de mensajes de texto.
1721. Entre las funcionalidades avanzadas se encuentra la posibilidad de mostrar contenidos de texto, contenidos web (enlaces, HTML, JavaScript,

etc.), gestionar datos binarios, como tonos de llamada, e intercambiar ficheros multimedia (audio, vídeo e imágenes), sobre todo en sus variantes avanzadas: EMS (*Enhanced Messaging Service*) y MMS (*Multimedia Message Service*).

1722. Debido a las diferentes vulnerabilidades asociadas al módulo de mensajería de texto existentes en el pasado en múltiples dispositivos móviles actuales, incluyendo Android [Ref.- 400], se recomienda que el usuario actúe con prudencia ante la lectura de nuevos mensajes de texto, especialmente los recibidos de fuentes desconocidas.

1723. Durante el año 2009 se publicaron diferentes vulnerabilidades de denegación de servicio (DoS) en el módulo SMS de dispositivos Android [Ref.- 400]. El proceso que gestiona los mensajes SMS en Android (SMSManager¹⁰⁷ o Provider) es una aplicación Java:

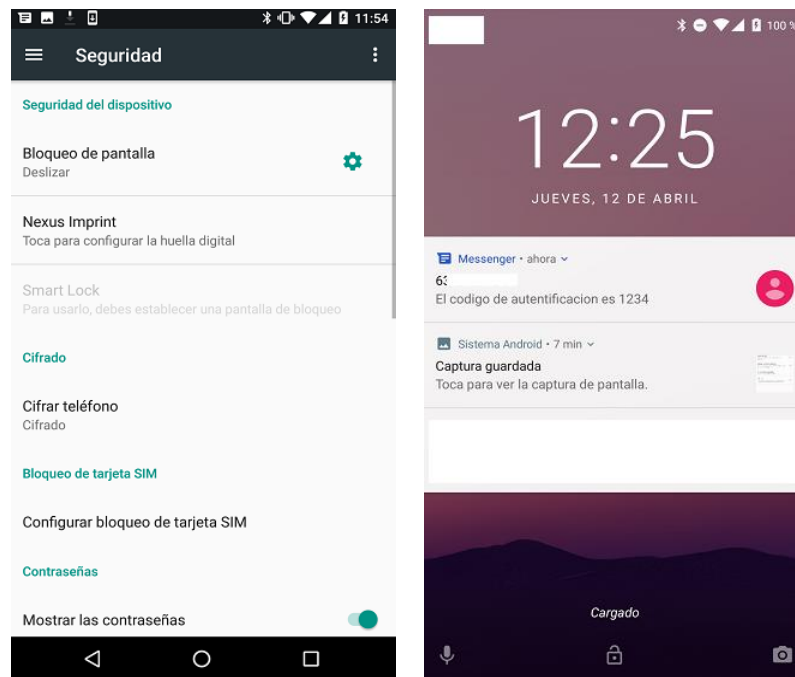


1724. Se recomienda no abrir ningún mensaje de texto no esperado o solicitado, práctica similar a la empleada para la gestión de correos electrónicos (o e-mails). A la hora de acceder y procesar los contenidos del buzón de entrada de mensajes de texto del dispositivo móvil, el usuario debería seguir las mejores prácticas de seguridad, por ejemplo:

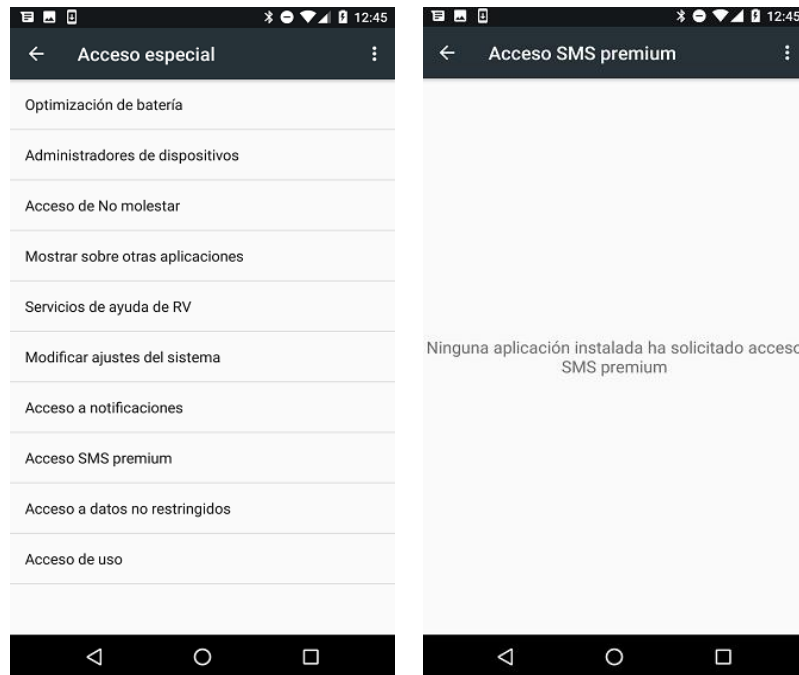
- No deberían abrirse mensajes de texto recibidos desde números desconocidos, ni ejecutar o permitir la instalación de contenidos adjuntos (configuraciones, binarios o multimedia) asociados a mensajes SMS/MMS no esperados.
- En su lugar, se debería proceder a borrar el mensaje directamente.

¹⁰⁷ <http://developer.android.com/reference/android/telephony/SmsManager.html>

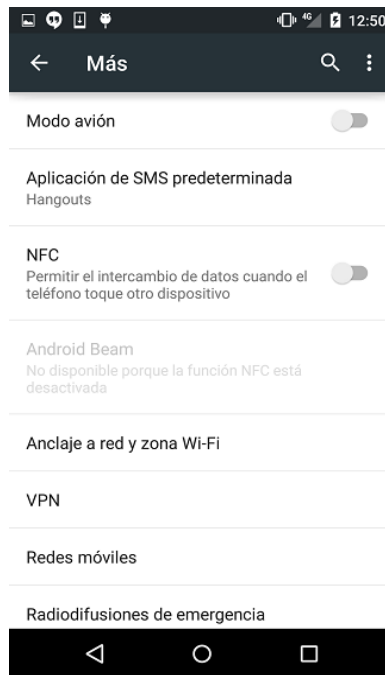
- En caso de acceder a los mensajes, debería desconfiarse de su contenido, y tener especial cuidado, por ejemplo, con la utilización de enlaces a sitios web externos.
1725. Adicionalmente, es necesario tener en cuenta que los mensajes de texto pueden contener información sensible y confidencial. Por ejemplo, al gestionar el buzón de voz de un usuario, los operadores de telefonía móvil pueden enviar todos los detalles de acceso al buzón de voz, incluido el PIN, mediante un mensaje de texto. El acceso a dicho mensaje por parte de un potencial atacante permitiría disponer de control completo del buzón de voz del usuario.
1726. Los nuevos mecanismos de autenticación de la banca online también emplean los mensajes de texto para el envío de contraseñas de un solo uso para la realización de operaciones bancarias críticas. El acceso a dicho mensaje por parte de un potencial atacante permitiría completar la operación, por ejemplo, una transferencia bancaria.
1727. Un punto a tener muy en cuenta es que, cuando el número de teléfono asociado a la tarjeta SIM del dispositivo móvil se emplea como segundo factor de autenticación para el acceso a una aplicación o a una cuenta asociada a un servicio (como la propia cuenta de usuario de Google), uno de los mecanismos utilizados para la verificación de identidad se basa en el envío mediante SMS de un código al dispositivo móvil, el cual deberá proporcionarse para validar el acceso. Por tanto, la protección del contenido de los SMS cobra aún mayor importancia (para más información sobre la configuración del segundo factor de autenticación, se recomienda consultar el apartado "Inicio de sesión y seguridad" de la "Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android" [Ref.- 407]).
1728. La recepción de mensajes SMS por parte del dispositivo móvil cuando la pantalla de bloqueo está bloqueada depende de la configuración seleccionada inicialmente por el usuario durante el proceso de configuración de un código de acceso respecto a las notificaciones. Si el dispositivo móvil no dispone de un código de acceso (opción desaconsejada desde el punto de vista de seguridad), las notificaciones relacionadas con la app "Mensajes" (o "Messenger", que es la app proporcionada por defecto para la gestión de SMS en Android 7.x) mostrarán todo el contenido de la notificación, salvo que las mismas se hayan bloqueado de forma global. Es decir, si no existe un mecanismo de bloqueo (o el único mecanismo activo es "deslizar"), no es posible filtrar la presentación de los contenidos de los SMS según su sensibilidad:



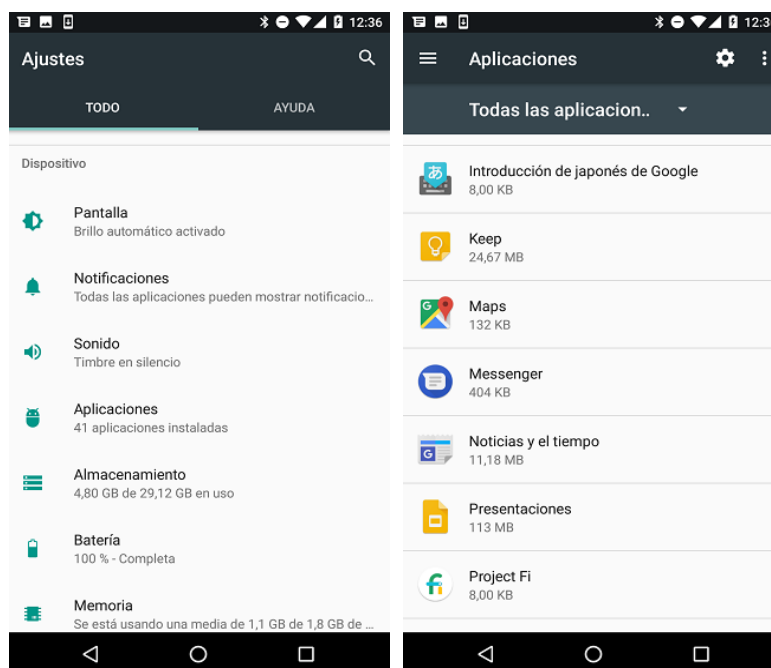
1729. Si la opción seleccionada fue "Mostrar todo el contenido de las notificaciones", opción más insegura, Android mostrará los contenidos del mensaje mediante una notificación en la pantalla de bloqueo (sirva como ejemplo la imagen superior derecha).
1730. Los ajustes de configuración de las notificaciones están disponibles bajo "Ajustes - [Dispositivo] Notificaciones", proporcionándose tres opciones para el escenario en el que el dispositivo está bloqueado: "No mostrar notificaciones" (opción más segura, pero que limita la funcionalidad de notificaciones en la pantalla de bloqueo), "Mostrar todo el contenido de las notificaciones" (opción más insegura) y "Ocultar notificaciones sensibles". Para más información, consultar el apartado "9.8. Notificaciones".
1731. Por otro lado, el envío de mensajes de texto (SMS) a servicios *Premium* ha sido una técnica muy comúnmente utilizada por software malicioso (*malware*) móvil para obtener beneficios económicos. Especialmente, esta técnica de obtención de financiación de manera ilegal ha sido empleada al infectar dispositivos móviles basados en Android, debido a las facilidades existentes para el envío de SMS *Premium* en esta plataforma móvil en versiones previas de Android y al coste asociado para el usuario por cada mensaje enviado (por ejemplo, un euro por SMS).
1732. Por este motivo, Android 4.2 introdujo nuevos mecanismos de protección a la hora de enviar mensajes SMS *Premium*, por ejemplo, por parte de una app, generándose un mensaje de notificación que debe ser aprobado por el usuario antes de proceder al envío del SMS al servicio *Premium*, con cargos adicionales [Ref.- 86].
1733. En Android 7.x es posible determinar si una aplicación ha solicitado permisos para enviar mensajes SMS Premium a través de "Ajustes - [Dispositivo] Aplicaciones - Engranaje de ajustes - [Configurar Aplicaciones] Opciones Avanzadas - Acceso especial":

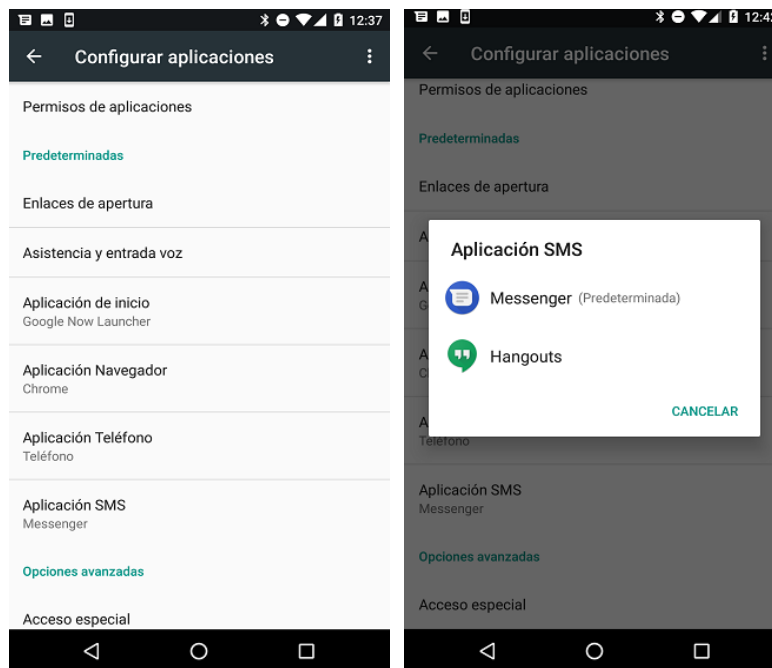


1734. No obstante, para evitar riesgos, se recomienda también contactar con el proveedor de servicios de telefonía móvil y solicitar que se habilite la restricción de envío de mensajes SMS Premium desde el número de teléfono asociado a la tarjeta SIM, salvo que se desee expresamente hacer uso de esta funcionalidad.
1735. Por otro lado, desde Android 4.4 (KitKat), la plataforma permite definir cuál es la app por defecto encargada de la gestión de mensajes SMS (*default SMS app*) [Ref.- 30], pudiendo haber únicamente una app gestionando la entrada de nuevos SMS (o MMS) al dispositivo móvil, y especialmente, disponiendo de capacidades de escritura en el proveedor de telefonía de Android.
1736. En la versión 5.x de Android, la app encargada de la gestión de mensajes SMS se definía a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Aplicación de SMS predeterminada":



1737. Sin embargo, en Android 7.x, la app responsable de la gestión de SMS se configura a través de "Ajustes - [Dispositivo] Aplicaciones - Engranaje de ajustes - [Configurar Aplicaciones] Predeterminadas - Aplicación SMS":





Nota: Existieron en el pasado soluciones propietarias de cifrado de mensajes de texto (que reemplazan o complementan a los SMS) y de mensajería en general (chat), como TextSecure (de código abierto), proporcionadas por terceras compañías a través de Google Play, que cifran los mensajes de texto tanto durante su almacenamiento en el dispositivo móvil como durante su transmisión a otros usuarios de la app TextSecure [Ref.- 33]. TextSecure hace uso del protocolo OTR (*Off The Record*), con mejoras¹⁰⁸, para cifrar los mensajes de texto. TextSecure puede ser empleado para el envío de mensajes cifrados entre terminales Android, o también con terminales basados en iOS mediante la app Signal¹⁰⁹, empleando las comunicaciones de datos (Wi-Fi o 2/3/4G) de los dispositivos móviles, o la infraestructura SMS de los operadores. El uso de la infraestructura de SMS asociada a los proveedores u operadores de telecomunicaciones para el envío de mensajes de texto puede revelar información en los metadatos de dichas conversaciones.

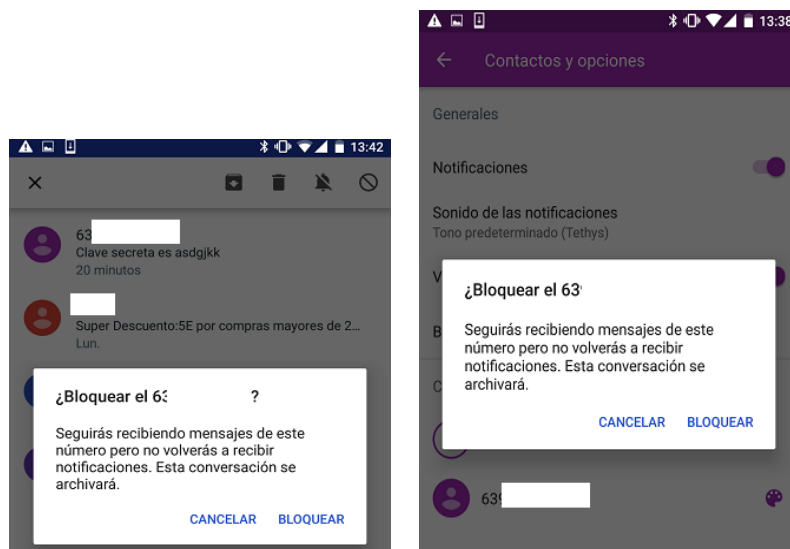
Actualmente, las soluciones y apps de mensajería (WhatsApp, Signal, Telegram, Line, Viber, etc.) han reemplazado de manera generalizada el uso de los mensajes SMS, haciendo una amplia utilización de diferentes mecanismos de cifrado, y en algunos casos como Signal [Ref.- 32], sustituyendo por completo apps previas, como TextSecure.

1738. La aplicación "Messenger" permite bloquear las notificaciones asociadas a los SMS procedentes de determinados números. Ello no implica que se dejen de recibir SMS de dichos números, sino que no se mostrarán al usuario. Para activar esta funcionalidad, se ha de acceder al interfaz de la aplicación, seleccionar el icono asociado al emisor del mensaje y pulsar el icono "prohibido" de la barra superior. También se puede seleccionar un mensaje

¹⁰⁸ <https://whispersystems.org/blog/simplifying-otr-deniability/>

¹⁰⁹ <https://github.com/WhisperSystems/Signal-iOS>

de dicho emisor y pulsar sobre "[...] - Contactos y opciones - [Generales] Bloquear el <número de teléfono>":



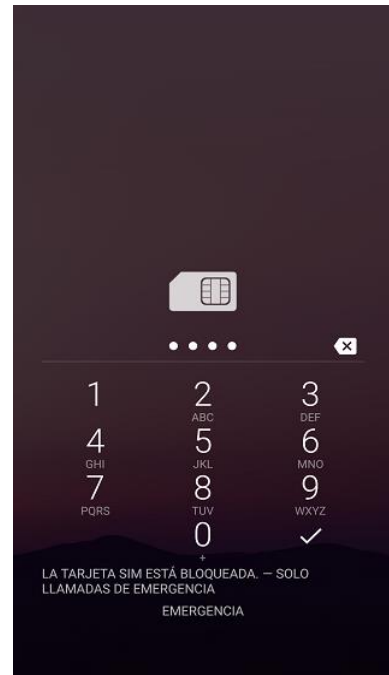
18. COMUNICACIONES MÓVILES: VOZ Y DATOS

18.1. DESACTIVACIÓN DE LAS COMUNICACIONES DE VOZ Y DATOS MÓVILES

1739. La principal recomendación de seguridad asociada a las comunicaciones de voz (y SMS) a través de las redes de telefonía móvil en dispositivos móviles es ser cauteloso a la hora de activar las distintas capacidades de telefonía del terminal. Así, por ejemplo, si se está conectado a una red Wi-Fi, se puede deshabilitar temporalmente el uso de datos móviles.

Nota: Durante el proceso de configuración inicial del dispositivo móvil empleado para la elaboración de la presente guía (activado automáticamente la primera vez que se enciende el terminal), el interfaz de telefonía móvil para comunicaciones de voz está habilitado, solicitando en primer lugar el PIN de la tarjeta SIM, si ésta está insertada. Se recomienda desactivar el interfaz de telefonía móvil tan pronto finalice el proceso de configuración inicial, para proceder a la aplicación de las medidas de seguridad seleccionadas.

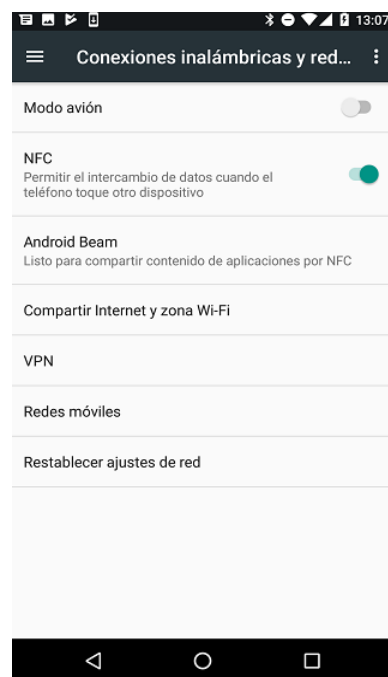
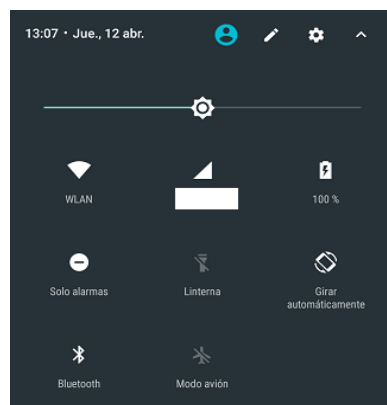
1740. En Android 7.x (al igual que ocurría en Android 6.x), y en concreto para el dispositivo móvil analizado, cada vez que se inserta una nueva tarjeta SIM se solicita al usuario el PIN asociado, no siendo necesario reiniciar el dispositivo para poder hacer uso de las capacidades de telefonía móvil (acceso a la red móvil). Al insertar una tarjeta SIM, Android solicitará al usuario el PIN asociado a ella; al extraer la tarjeta SIM, se mostrará un mensaje en la barra de notificaciones que informa de que la misma no está disponible. En versiones previas de Android, y en otros modelos de terminal, puede ser necesario reiniciar el dispositivo móvil cada vez que se inserte o extraiga la tarjeta SIM:



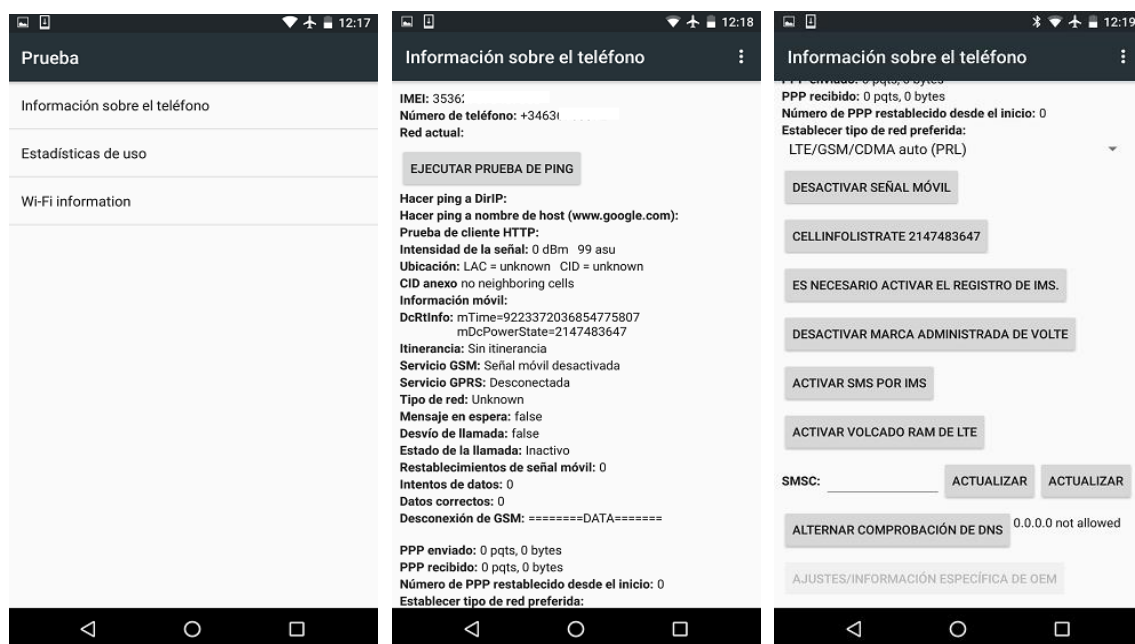
1741.

1742. Android no permite a través del interfaz de usuario estándar deshabilitar de forma individualizada las capacidades de telefonía móvil (voz y SMS); la única opción para ello es activar el "Modo avión", bien a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Más..." o del menú de ajustes rápidos.

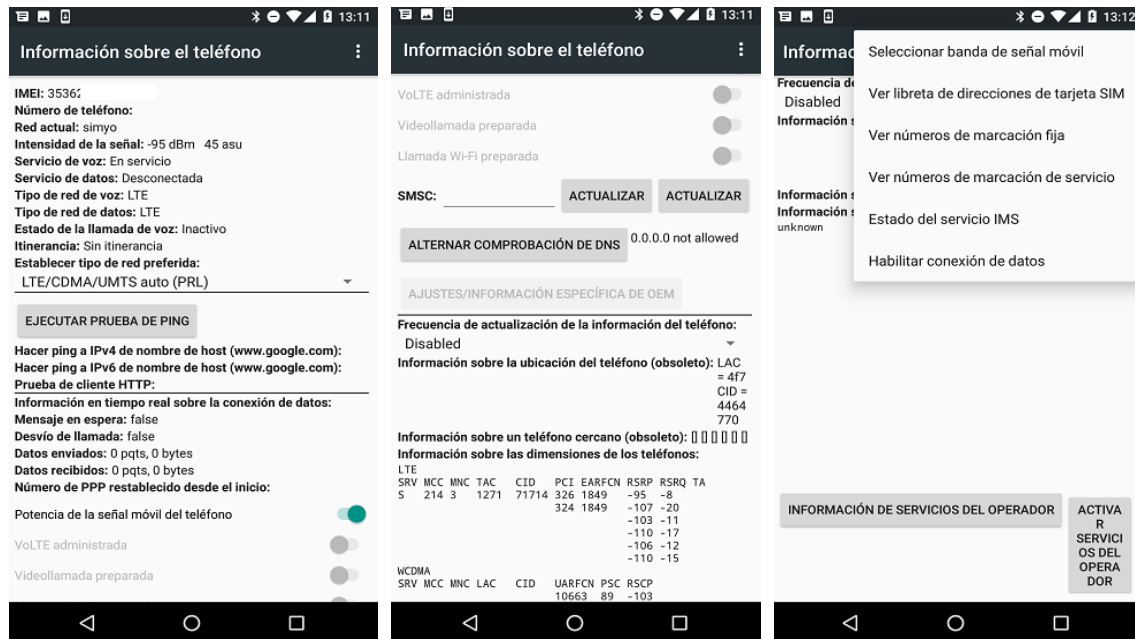
1743. Al activar el modo avión en Android, tanto la conectividad a redes Wi-Fi como la conectividad de Bluetooth puede ser habilitada posteriormente y de forma independiente, pero no es posible activar la conectividad de telefonía móvil en ese modo, ni tampoco la conectividad móvil 2/3/4G, directamente ligada al modo avión:



1744. Debido a este comportamiento, no resulta posible desactivar las capacidades de voz únicamente, ni siquiera en el caso de tabletas que emplean el SIM exclusivamente para conexión de datos.
1745. Existe un menú de configuración avanzada denominado "Prueba" (o *testing*), al cual se accede tras marcar el siguiente código en el menú de marcación telefónica: `##4636##` (o `##info##`).
1746. En versiones anteriores de Android (hasta la 6.x inclusive), esta funcionalidad permitía independizar la desconexión de los diferentes interfaces de comunicaciones inalámbricos, es decir, Bluetooth, Wi-Fi y telefonía móvil 2/3/4G.
1747. Por ejemplo, era posible desactivar la señal móvil accediendo al menú "Prueba - Información sobre el teléfono", e independientemente del estado de activación de los interfaces Bluetooth y Wi-Fi, se podía acceder a un menú que permitía desconectar únicamente el interfaz de telefonía móvil 2/3/4G: así, el botón "Desactivar señal móvil" que aparece al descender en la pantalla, desactivará únicamente y por completo (voz y datos) el interfaz de radio de telefonía móvil, pudiendo permanecer los interfaces de Bluetooth y Wi-Fi activos, tal y como denotan los iconos de la barra superior de estado (las imágenes que se muestran a continuación corresponden a un dispositivo Nexus 5X con Android 6.0.1):

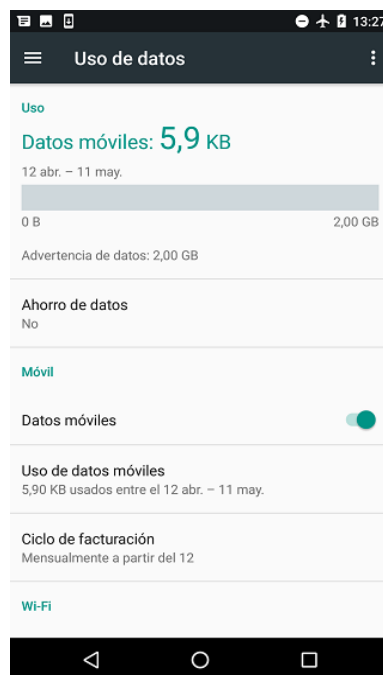


1748. Este cambio de configuración no prevalecía tras reiniciar el dispositivo móvil, donde de nuevo el interfaz de radio de telefonía móvil era activado automáticamente tras introducir el PIN de la tarjeta SIM.
1749. En Android 7.x el menú de prueba, si bien permite consultar la información sobre el teléfono a nivel de telefonía móvil, no permite ya desactivar la señal, como muestran las siguientes capturas de pantalla:



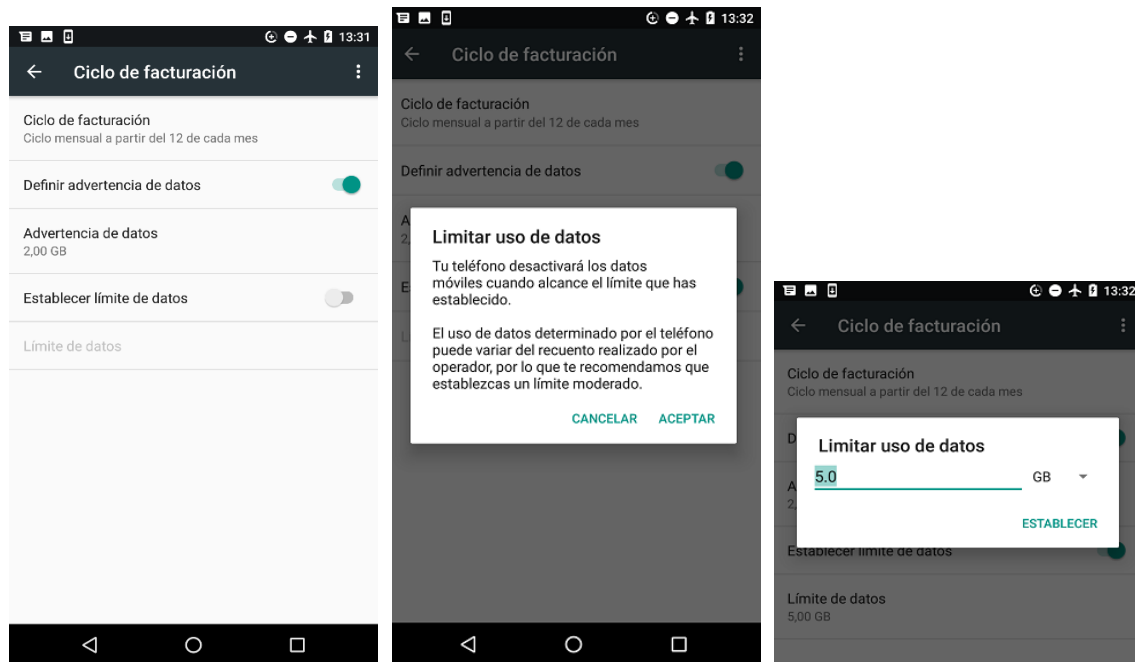
1750. No obstante, la principal recomendación de seguridad asociada a las comunicaciones de datos a través de las infraestructuras de telefonía móvil en dispositivos móviles es no activar las capacidades de transmisión y recepción de datos salvo en el caso en el que se esté haciendo uso de éstas, evitando así la posibilidad de ataques sobre el hardware del interfaz, el driver, la pila de comunicaciones móviles y cualquiera de los servicios y apps disponibles a través de esa red y la conexión a Internet.

1751. Las conexiones de datos a través de la red de telefonía móvil (2/3/4G), GPRS, UMTS o LTE, pueden ser deshabilitadas a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Uso de datos" y desactivando el interruptor "Datos móviles".



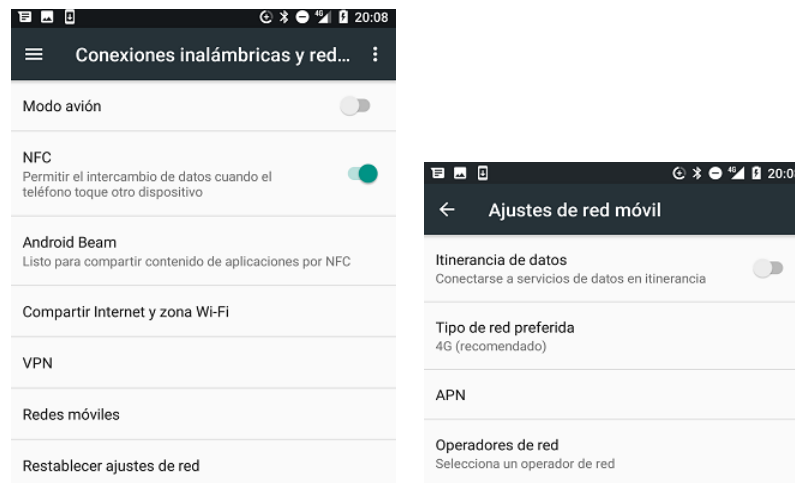
1752. También es posible establecer un límite de datos, de forma que el sistema Android desactive automáticamente los datos móviles cuando se

alcance el umbral definido, impidiendo así que las aplicaciones puedan consumir datos. Para configurar dicho límite, se dispone de la opción "Uso de datos móviles - <rueda de ajustes> [Ciclo de facturación] - Establecer límite de datos" y, dentro de esa misma pantalla, marcar "Límite de datos". A diferencia de lo que ocurría hasta Android 6.x, en lugar de configurar el límite de datos desplazando la barra horizontal del gráfico, el límite se fija de forma numérica. Además, se permite establecer un parámetro "Advertencia de datos" para que el dispositivo alerte de que se ha llegado a un umbral de consumo de datos definido por el usuario:



1753. Android 7.x dispone de una opción que permite al usuario crear una lista blanca de apps autorizadas a las que sí se permitirá hacer uso de las comunicaciones de datos móviles cuando estén ejecutando de fondo incluso aunque el servicio de ahorro de datos "Data Saver" (descrito al final de este apartado) se encuentre activo.

1754. La opción "Itinerancia de datos" (o *roaming*), disponible a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Redes móviles", permite configurar si se desea hacer uso de las capacidades de datos de telefonía móvil del dispositivo cuando se viaja al extranjero y el terminal se encuentra en una red de otro operador al que puede asociarse su tarjeta SIM:



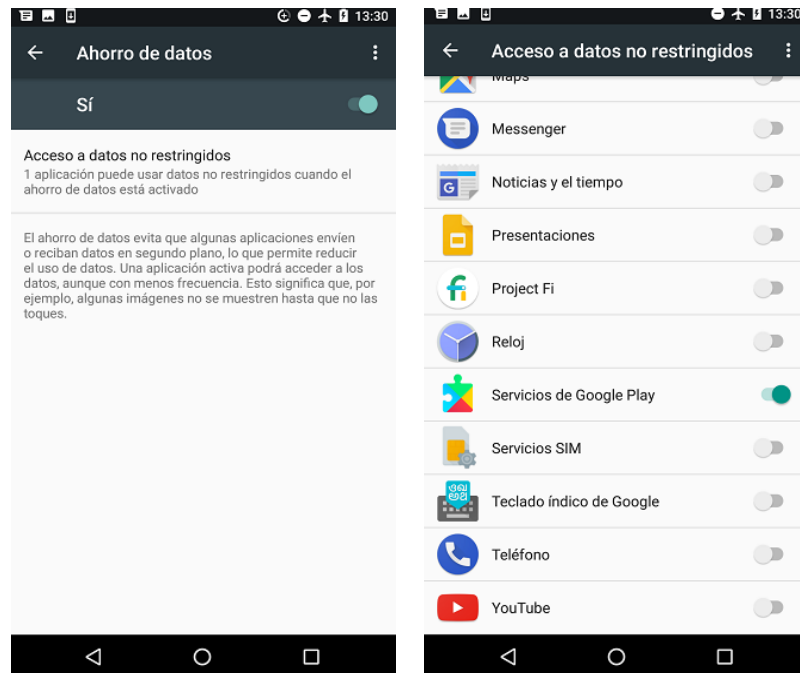
Nota: Además de la seguridad, una de las preocupaciones de los usuarios de dispositivos móviles sin tarifa plana de conexión de datos son los cargos asociados al uso de las comunicaciones de datos a través de las redes de telefonía móvil (2/3/4G), especialmente cuando el contrato tiene gastos asociados por la cantidad de datos transmitidos y/o por tiempo o número de conexiones establecidas, o se viaja internacionalmente (itinerancia o *roaming*). Múltiples apps en Android pueden iniciar este tipo de conexiones de datos en diferentes momentos del día.

- 1755. Por defecto en Android (configuración de fábrica) el interfaz de datos de telefonía móvil está habilitado (con los APNs que han sido definidos por la tarjeta SIM empleada).
- 1756. Android mantiene el estado del interfaz de telefonía de datos tras reiniciar el dispositivo móvil, es decir, si el interfaz de telefonía de datos estaba activo al apagar el terminal, al encender el dispositivo móvil e introducir el PIN de la tarjeta SIM, seguirá activo.
- 1757. Igualmente, al salir del modo avión el estado del interfaz de datos de telefonía móvil será restaurado, y se activará en el caso de haber estado activo previamente (antes de activar el modo avión).
- 1758. Tras desactivar la conexión de datos, si alguna app requiere disponer de conectividad (por ejemplo, acceso a la web desde el navegador web existente por defecto) la conexión de telefonía de datos no será activada automáticamente.

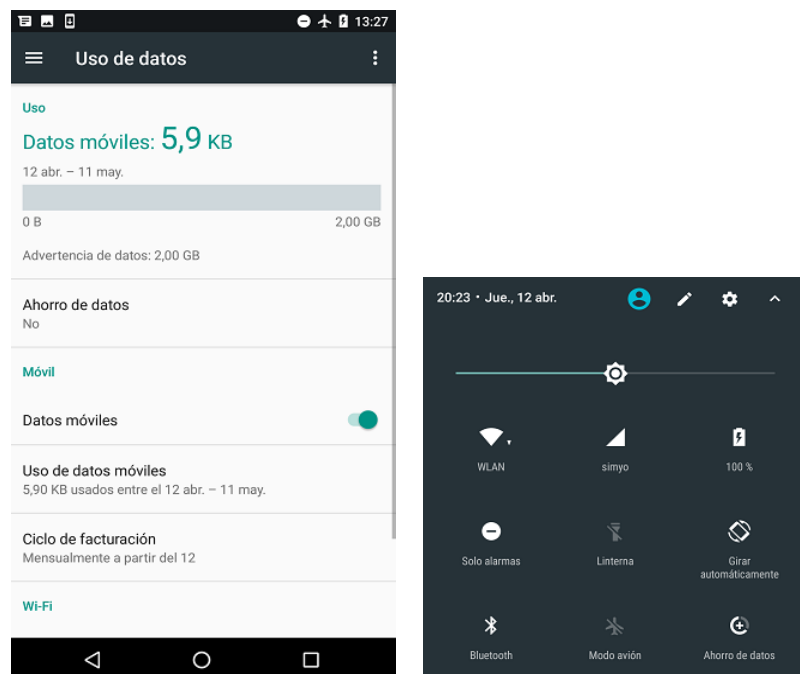
18.1.1. DATA SAVER

- 1759. Una de las novedades introducidas por Android 7.x es la denominada "Data Saver" (ahorro de datos), un nuevo modo por el cual las aplicaciones no pueden consumir datos móviles mientras están ejecutando en segundo plano.
- 1760. Sin embargo, se permite al usuario establecer una lista blanca de aplicaciones que podrán saltarse este ajuste y continuar con el consumo de datos móviles.
- 1761. Esta lista blanca se configura a través del menú "Ajustes - Uso de datos - Ahorro de datos - Acceso a datos no restringidos". Además, en esta pantalla

se puede ver si alguna app tiene este permiso. Por defecto, la app "Servicios de Google Play" se incluye en esta lista:



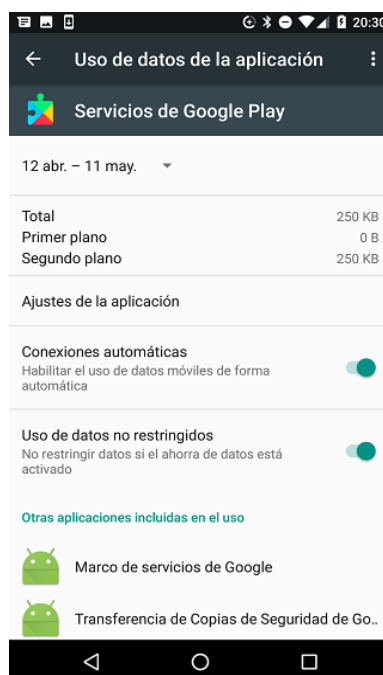
1762. El servicio "Data Saver" se puede activar, bien mediante el menú "Ajustes - [Conexiones inalámbricas y redes] Uso de datos - Ahorro de datos", como también mediante el menú de ajustes rápidos con la opción "Ahorro de datos":



1763. Cuando el modo de ahorro de datos está activo, en la barra superior de estado se mostrará un símbolo "+" rodeado de un círculo incompleto. Además, en la parte superior del menú de ajustes aparecerá un mensaje informativo:



1764. Para poder conocer los datos móviles que consume una determinada app cuando ejecuta en segundo plano, se accederá a "Ajustes - [Dispositivo] Aplicaciones - [Nombre de la app] Uso de datos":



1765. Para dotar a las apps de capacidad para saber si el modo de ahorro de datos está activo, Google ha añadido funcionalidad a la API `ConnectivityManager`¹¹⁰, incluyendo la posibilidad de que la app sepa si se ha añadido a la lista blanca o no.
1766. Además, a través de ADB se pueden consultar los ajustes de ahorro de datos, incluyendo los UIDs de las apps que se encuentran en la lista blanca:

¹¹⁰ <https://developer.android.com/reference/android/net/ConnectivityManager.html>

```

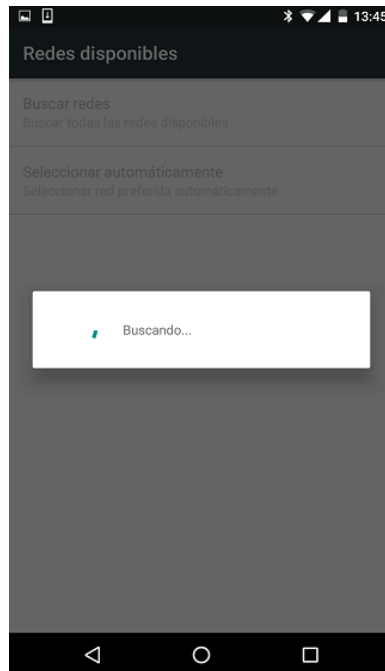
C:\> adb shell dumpsys netpolicy
System ready: true
Restrict background: true
Restrict power: false
Device idle: false
Network policies:
  NetworkPolicy[NetworkTemplate: matchRule=MOBILE_ALL, subscriberId=214035...,
matchSubscriberIds=[214035...]]: cycleDay=12, cycleTimezone=Europe/Madrid,
warningBytes=2147483648, limitBytes=5368709120, lastWarningSnooze=-1,
lastLimitSnooze=-1, metered=true, inferred=false
Metered ifaces: {rmnet_data0}
Policy for UIDs:
Power save whitelist (except idle) app ids:
  UID=10005: true
  UID=10011: true
  UID=10014: true
  UID=10020: true
  UID=10033: true
  UID=10086: true
Power save whitelist app ids:
  UID=10005: true
  UID=10011: true
  UID=10014: true
  UID=10020: true
  UID=10086: true
Restrict background whitelist uids:
  UID=10005
  UID=10014
  UID=10020
Default restrict background whitelist uids:
  UID=10005
  UID=10014
  UID=10020
Status for all known UIDs:
  UID=1001 state=0 (fg) rules=0 (NONE)
  UID=1002 state=3 (fg svc) rules=0 (NONE)
  UID=1027 state=0 (fg) rules=0 (NONE)
  UID=10012 state=16 (bg) rules=0 (NONE)
  UID=10014 state=10 (bg) rules=1 (ALLOW_METERED)
  UID=10020 state=3 (fg svc) rules=1 (ALLOW_METERED)
  UID=10023 state=3 (fg svc) rules=0 (NONE)
  UID=10026 state=16 (bg) rules=0 (NONE)
  UID=10027 state=16 (bg) rules=0 (NONE)
  UID=10033 state=10 (bg) rules=0 (NONE)
  UID=10036 state=10 (bg) rules=0 (NONE)
  UID=10040 state=0 (fg) rules=0 (NONE)
  UID=10043 state=3 (fg svc) rules=2 (TEMPORARY_ALLOW_METERED)
  UID=10073 state=7 (bg) rules=0 (NONE)
  UID=10079 state=16 (bg) rules=0 (NONE)
  UID=10082 state=10 (bg) rules=0 (NONE)
  UID=10085 state=16 (bg) rules=0 (NONE)
Status for just UIDs with rules:
  UID=10014 rules=1 (ALLOW_METERED)
  UID=10020 rules=1 (ALLOW_METERED)
  UID=10043 rules=2 (TEMPORARY_ALLOW_METERED)

```

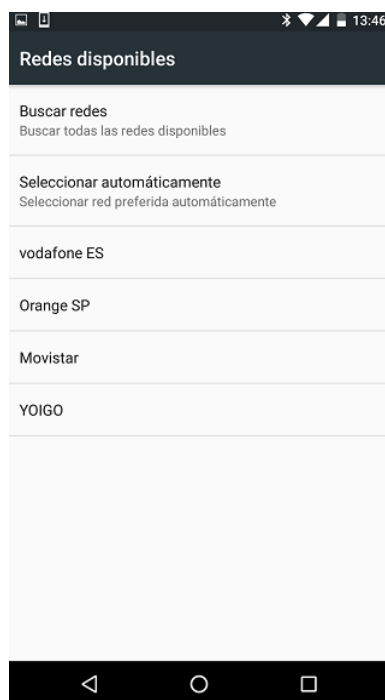
18.2. CONFIGURACIÓN Y SELECCIÓN DE LA RED (VOZ) DE TELEFONÍA MÓVIL

1767. Android permite al usuario utilizar la red de telefonía móvil configurada automáticamente a través de la tarjeta SIM, así como seleccionar de forma manual el operador de telecomunicaciones móviles (voz y SMS).
1768. Desde el menú de configuración del teléfono ("Ajustes [Conexiones Inalámbricas y Redes] - Más... - Redes móviles") y a través de la opción

"Operadores de red", es posible acceder a las opciones para seleccionar el operador de red, tanto de forma automática como manual:

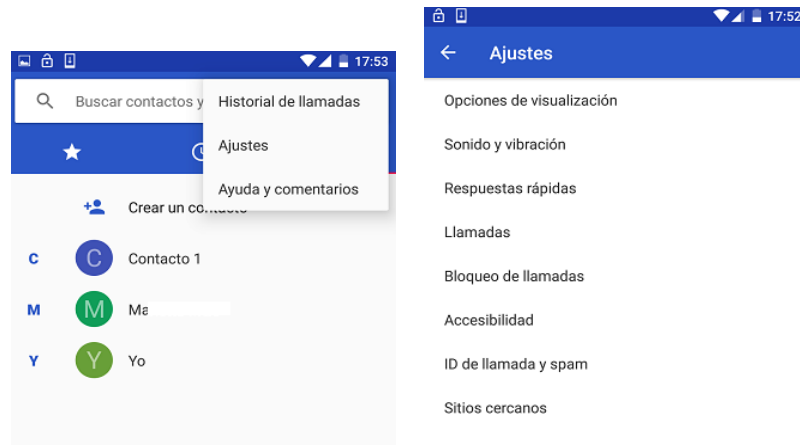


1769. Android llevará a cabo una búsqueda automática de las redes disponibles en la ubicación actual del dispositivo móvil, y mostrará al usuario la lista de operadores de telecomunicaciones identificados:

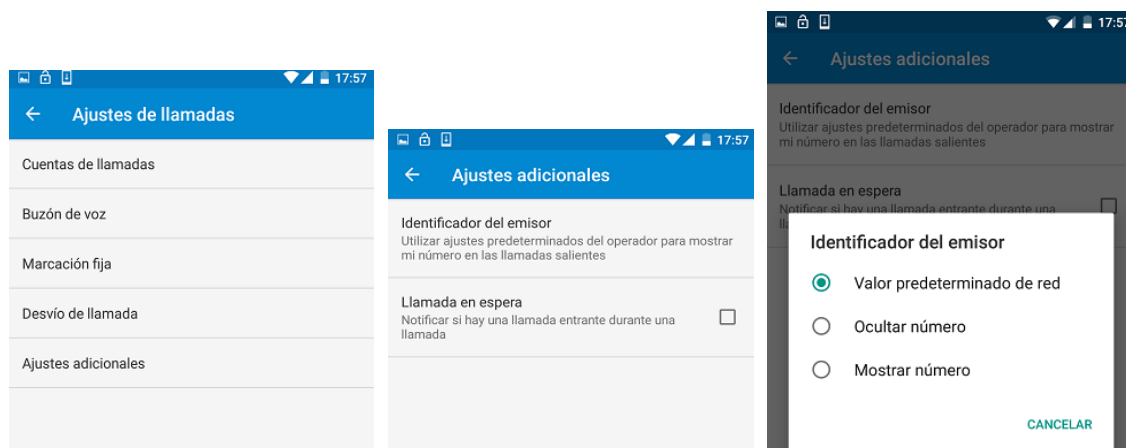


1770. La selección de red permite definir si la red de telefonía será seleccionada de forma automática (opción por defecto) por el dispositivo móvil en función de la tarjeta SIM y de la intensidad de la señal, o si el usuario puede definir de forma manual la red a la que desea conectarse de la lista de redes disponibles.

1771. Adicionalmente, Android dispone en sus versiones más recientes de otras capacidades y ajustes de configuración asociados a la app del teléfono¹¹¹ que pueden tener implicaciones desde el punto de vista de seguridad. Para acceder a estas capacidades, se abrirá la app "Teléfono" y se pulsará la opción "... " de la esquina superior derecha seguida de "Ajustes":



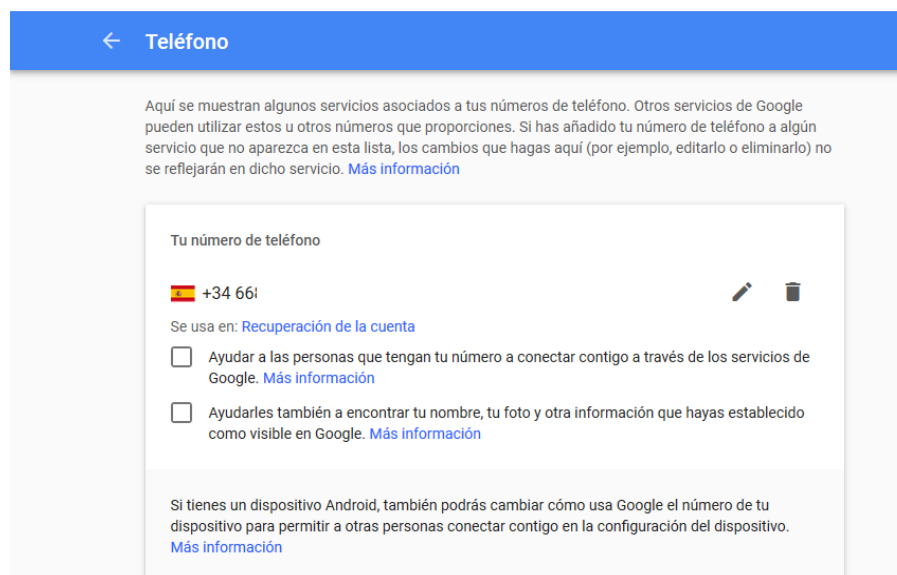
1772. Con el objetivo de proteger la privacidad del usuario, el módulo de telefonía de Android, en concreto la app del teléfono, permite al usuario definir en qué tipo de llamadas se enviará la información de identificación como autor de la llamada: "Valor predeterminado de red" (opción por defecto), "Ocultar número" (siempre se ocultará) o "Mostrar número" (siempre se mostrará para todas las llamadas realizadas).
1773. Desde la app del teléfono, mediante el menú de opciones avanzadas disponible en la parte superior derecha, a través del menú "Ajustes - Llamadas - Ajustes adicionales", "Identificador del emisor", es posible configurar la opción deseada:



Nota: El menú para acceder al ajuste de identificador del emisor en cada dispositivo móvil depende de la versión de la app "Teléfono", por lo que puede no coincidir con el descrito en la descripción anterior. La versión de la app "Teléfono" empleada para la elaboración de la presente guía es la 8.1.146524971.

¹¹¹ <https://support.google.com/nexus/answer/2895524>

1774. Desde la versión Android 4.4 (KitKat), Google ha implementado la identificación de llamadas inteligente, también denominada identificación de llamada de Google [Ref.- 148]. Cuando se recibe (o se realiza) una llamada de alguien que no está en la agenda o lista de contactos, el teléfono buscará coincidencias para las personas en base a la información disponible en su cuenta de usuario de Google (desde el año 2014¹¹²) y para las empresas que tengan una ficha local en Google My Business.
1775. En el caso de compañías y organizaciones que hagan uso de cuentas de trabajo, o de centros educativos, el servicio tratará de buscar coincidencias en los directorios propios para ver si contienen información sobre dichos contactos.
1776. La visibilidad del número de teléfono propio para otras personas puede ser modificada a través de los ajustes de la cuenta de usuario de Google, en concreto, desde el siguiente enlace "<https://myaccount.google.com/phone?pli=1>", mediante la opción disponible en la parte inferior:

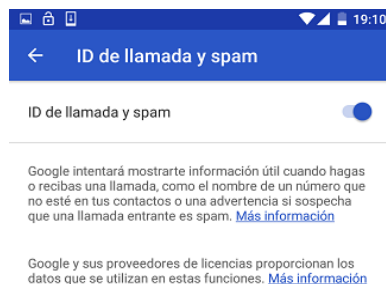


1777. La modificación del número de teléfono asociado a la cuenta de usuario de Google afecta a los servicios de Google listados en la página web anterior. Debe tenerse en cuenta que existen numerosos servicios que pueden hacer uso del número de teléfono del usuario¹¹³.
1778. La visibilidad del número de teléfono en la cuenta de usuario de Google está habilitada por defecto, por lo que se recomienda deshabilitarla si no se desea compartir el número de teléfono propio con otros usuarios. Si se deja habilitada, otros usuarios podrán enviar un mensaje (no sólo a través de aplicaciones como Hangouts que usan el número de teléfono, sino también mensajes de correo electrónico), o consultar el nombre o la foto del perfil al establecer una llamada.

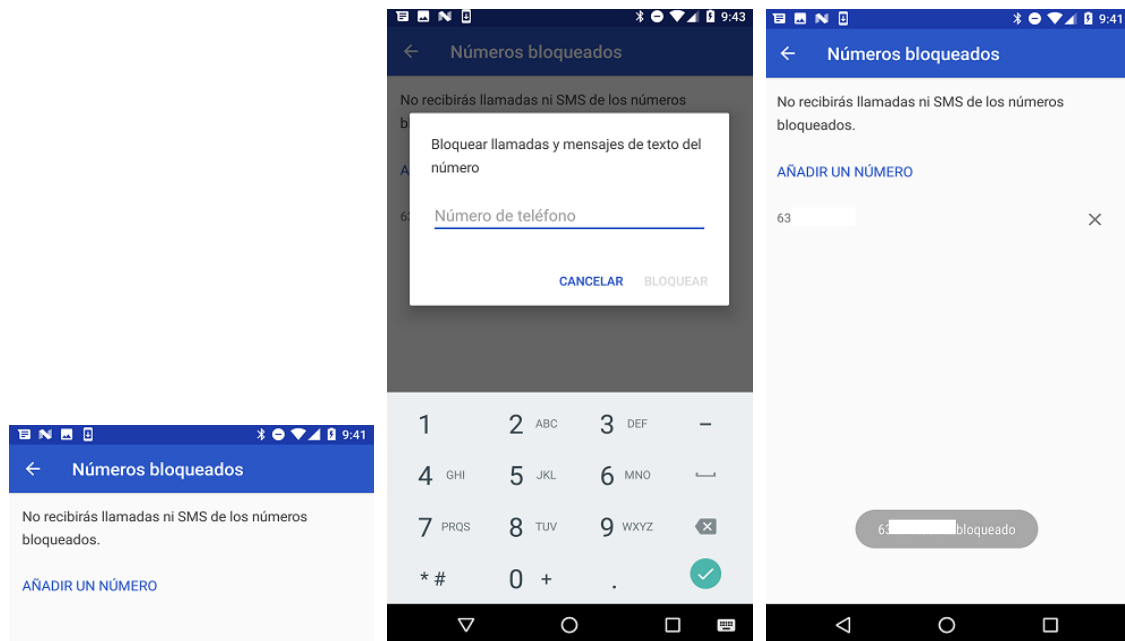
¹¹² <https://plus.google.com/photos/+android/albums/5942570061497141729>

¹¹³ <https://support.google.com/accounts/answer/3463280?p=modifynumber>

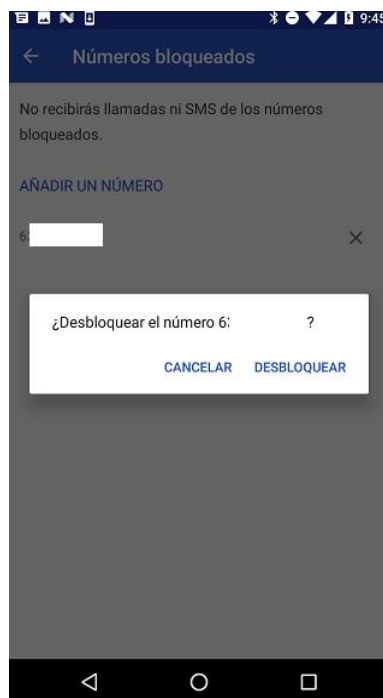
1779. Adicionalmente, existe una opción que permite que, al recibir o cursar una llamada asociada a un número que no forma parte de los contactos, se intente ofrecer información sobre el otro interlocutor. Esta funcionalidad está disponible en Android desde la app del teléfono, mediante el menú de opciones avanzadas disponible en la parte superior derecha, a través del menú "... - Ajustes - ID de llamada y spam" (opción que por defecto está habilitada) [Ref.- 147]:



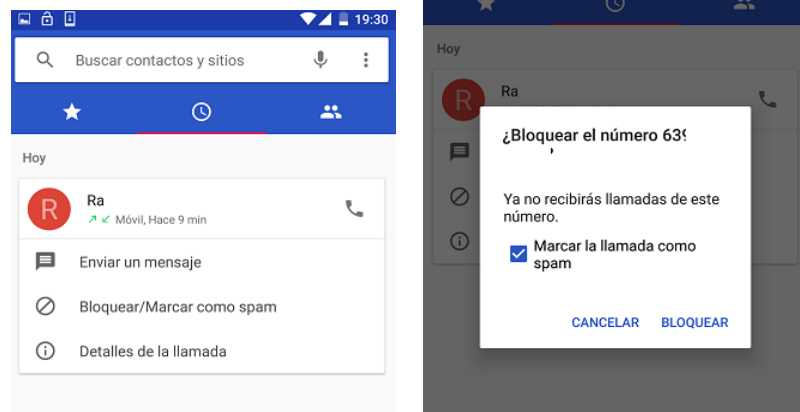
1780. Los identificadores de llamada de Google no se añaden automáticamente a la lista de contactos. Desde el punto de vista de seguridad, si no se desea hacer uso de esta funcionalidad, se recomienda deshabilitarla.
1781. En resumen: por un lado, desde la configuración del número de teléfono de la cuenta de usuario de Google, el propietario del dispositivo móvil puede definir si desea que otros usuarios tengan capacidad para localizarle en base al número de teléfono propio; por otro lado, desde los ajustes de la app "Teléfono" en Android es posible configurar si se desea localizar a otros usuarios (que lo permitan) a través de la funcionalidad del identificador de llamada de Google.
1782. Aunque en algunas versiones de la app "Teléfono" no existe una opción directa para bloquear números de teléfono de los cuales no se desea recibir llamadas, en la versión empleada en la elaboración de la presente guía sí está disponible desde el menú "Ajustes de la app Teléfono - Bloqueo de llamadas".



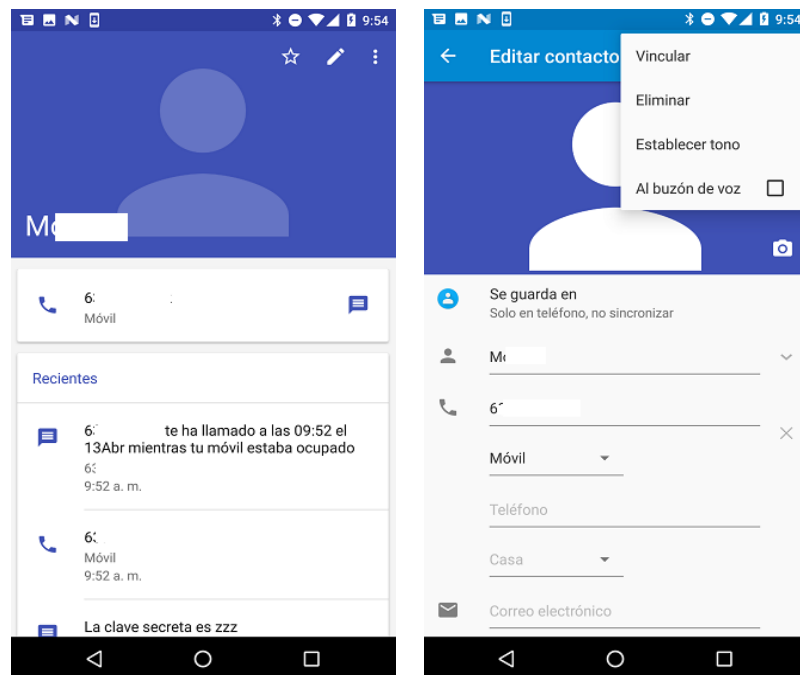
1783. Para dejar de bloquear un número previamente bloqueado, basta con pinchar sobre la "X" situada a la derecha del número y seleccionar "Desbloquear":



1784. Otra opción para bloquear un número es seleccionarlo desde la lista de contactos con una pulsación larga y marcar "Bloquear/Marcar como spam":

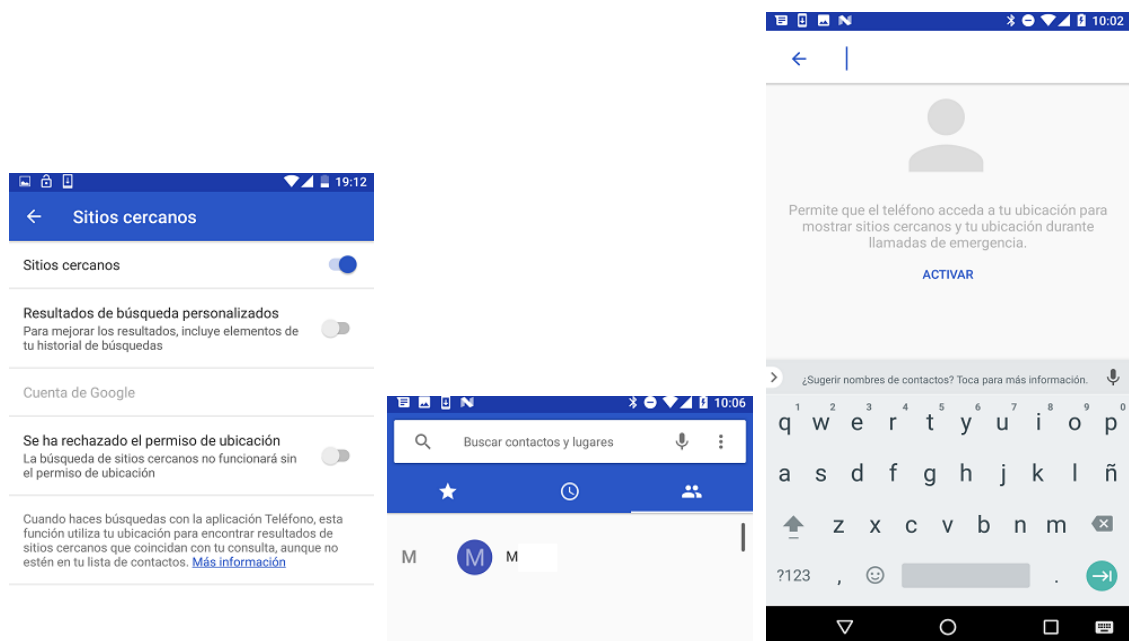


1785. Adicionalmente, se puede optar por redirigir las llamadas de un determinado contacto al buzón de voz de forma automática. Para ello, se ha de seleccionar el contacto y pulsar sobre el símbolo de lápiz de su parte superior para editar el contacto; en la ventana de edición que se abre, se ha de pulsar sobre los tres puntos "(...)" que representan los ajustes en la parte superior derecha de la app, seguido de la opción "Al buzón de voz". De este modo, las llamadas procedentes de ese número no se cursarán hacia el dispositivo móvil. Sin embargo, sí se recibirán los SMS desde dicho número.

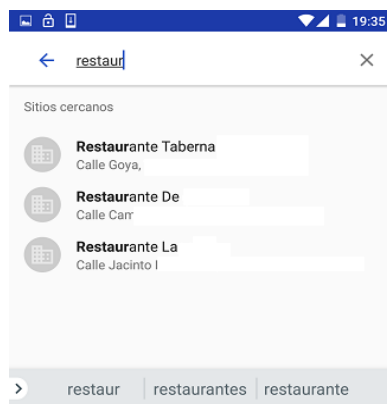


1786. Android 7.x dispone de capacidades en el teléfono para utilizar la ubicación del dispositivo móvil con el fin de encontrar sitios cercanos que coincidan con las búsquedas realizadas mediante la opción "Buscar contactos y lugares" de la app Teléfono, aunque la información proporcionada no esté disponible en la agenda o lista de contactos, actuando de guía telefónica.
1787. Esta opción (configurable a través del interruptor "Sitios cercanos" del menú de ajustes de la app Teléfono y habilitada por defecto) requiere que se

haya concedido a la app Teléfono el permiso de ubicación. En caso contrario, aunque la opción "Sitios cercanos" esté marcada, al intentar realizar una búsqueda el dispositivo mostrará una pantalla solicitando la activación del permiso de ubicación:



1788. La opción "Se ha rechazado el permiso de ubicación" aparecerá desmarcada si se denegó el permiso de ubicación a la app "Teléfono". Si se activa este interruptor, se abrirá un menú que solicitará de nuevo al usuario el permiso de ubicación para dicha app.
1789. Además, es preciso que el servicio de ubicación esté activo en el dispositivo móvil a nivel de los ajustes generales (menú "Ajustes - [Personal] Ubicación").
1790. Cuando se realiza una búsqueda en la agenda o lista de contactos desde la app "Teléfono", el teléfono buscará información relacionada de sitios cercanos y la mostrará en la parte superior. Para ello es necesario que estén activos los servicios de localización en Android:



1791. Los teléfonos sugeridos para sitios cercanos no se añaden automáticamente a la lista de contactos, aunque sí se pueden guardar posteriormente de forma manual, mediante la opción "Añadir a contactos".

1792. De nuevo, desde el punto de vista de seguridad, es preferible no hacer uso de esta funcionalidad, por lo que se recomienda deshabilitarla.

Nota: Android 4.4 (KitKat) y versiones posteriores permiten la realización de llamadas por Internet (VoIP, *Voice over IP*) mediante el protocolo SIP (*Session Initiation Protocol*) de señalización si el proveedor de telefonía ofrece dichas capacidades y el dispositivo móvil está conectado a una red Wi-Fi¹¹⁴.

Existen soluciones propietarias de cifrado extremo a extremo de las comunicaciones y llamadas de voz, como RedPhone (de código abierto), proporcionadas por terceras compañías a través de Google Play [Ref.- 29]. RedPhone hace uso del protocolo ZRTP para negociar las claves de cifrado empleadas en el protocolo SRTP, utilizado para cifrar las comunicaciones de Voz sobre IP (VoIP) entre dos terminales Android (o también con terminales basados en iOS mediante la app Signal¹¹⁵), haciendo uso de redes Wi-Fi o de la infraestructura de datos de los operadores de telefonía móvil, y empleando comunicaciones de datos cifradas mediante TLS con los servidores de RedPhone (para el protocolo de señalización) y mensajes SMS cifrados para el establecimiento de las llamadas.

Actualmente, las soluciones y apps de mensajería (WhatsApp, Signal, Telegram, Line, Viber, etc.) han incorporado capacidades para la realización de llamadas de voz en tiempo real mediante VoIP, haciendo una amplia utilización de diferentes mecanismos de cifrado, y en algunos casos como Signal [Ref.- 32], sustituyendo apps previas como RedPhone.

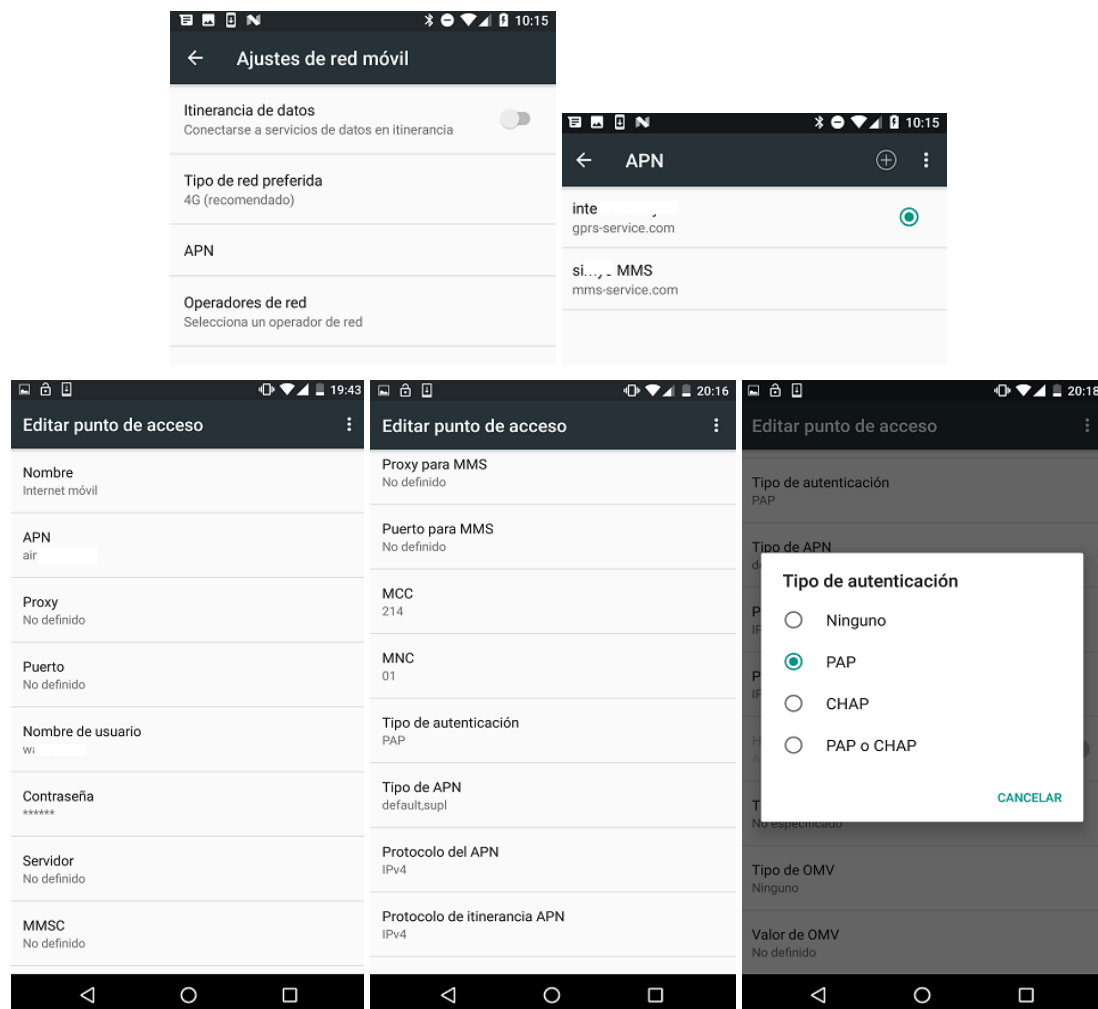
18.3. CONFIGURACIÓN DE LA RED DE DATOS MÓVILES

1793. Android permite al usuario, al igual que para la red de telefonía móvil (SMS y voz), utilizar la red de datos configurada automáticamente a través de la tarjeta SIM, así como seleccionar de forma manual tanto el operador de telecomunicaciones móviles como el APN (*Access Point Name*) para las conexiones de datos.
1794. En dispositivos móviles Android como el empleado para la elaboración de la presente guía, al introducir la tarjeta SIM en el dispositivo móvil se lleva a cabo automáticamente el proceso de configuración de las opciones de conectividad de datos asociadas a las redes de telefonía móvil, incluyendo los APNs y su configuración: nombre, APN, proxy y puerto, credenciales (usuario y contraseña), servidor, dispositivos de mensajes multimedia (MMS: MMSC, proxy y puerto MMS), MCC, MNC, tipo de autenticación y tipo de APN.
1795. En función de la tarjeta SIM empleada y de la configuración de datos que ésta tenga asociada, se mostrarán automáticamente los ajustes de conexión disponibles en la lista de APNs, o ningún APN si no dispone de los datos correspondientes (en cuyo caso habrá que añadirlo manualmente).

¹¹⁴ <https://support.google.com/nexus/answer/2811843>

¹¹⁵ <https://github.com/WhisperSystems/Signal-iOS>

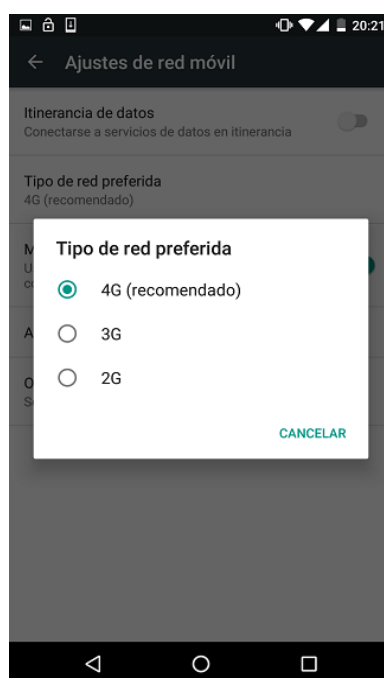
1796. Desde el menú de "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Redes móviles", y a través de la opción "APN", es posible acceder a las opciones para seleccionar el APN empleado para las conexiones de datos. Mediante la selección del APN de la lista es posible editar la configuración de los APNs existentes, y a través de los botones de la parte superior derecha, es posible añadir nuevos APNs o restablecer la configuración inicial:



1797. Al definir un nuevo APN deben emplearse los valores proporcionados por el operador de telecomunicaciones. Desde el punto de vista de seguridad, y si el operador admite diferentes protocolos de autenticación, se recomienda seleccionar autenticación mediante CHAP, dentro de las cuatro alternativas disponibles en la opción "Tipo de autenticación" (Ninguno, PAP, CHAP, y "PAP o CHAP"). Por defecto, el tipo de autenticación no está definido en el perfil del APN, como se ilustra en la imagen superior derecha:
1798. En el escenario de autoconfiguración existente por defecto, desde el punto de vista de seguridad, se recomienda acceder a la configuración de datos de telefonía móvil y, en concreto, a la lista de APNs para adecuar la configuración a la deseada por el usuario cada vez que se cambia la tarjeta SIM.

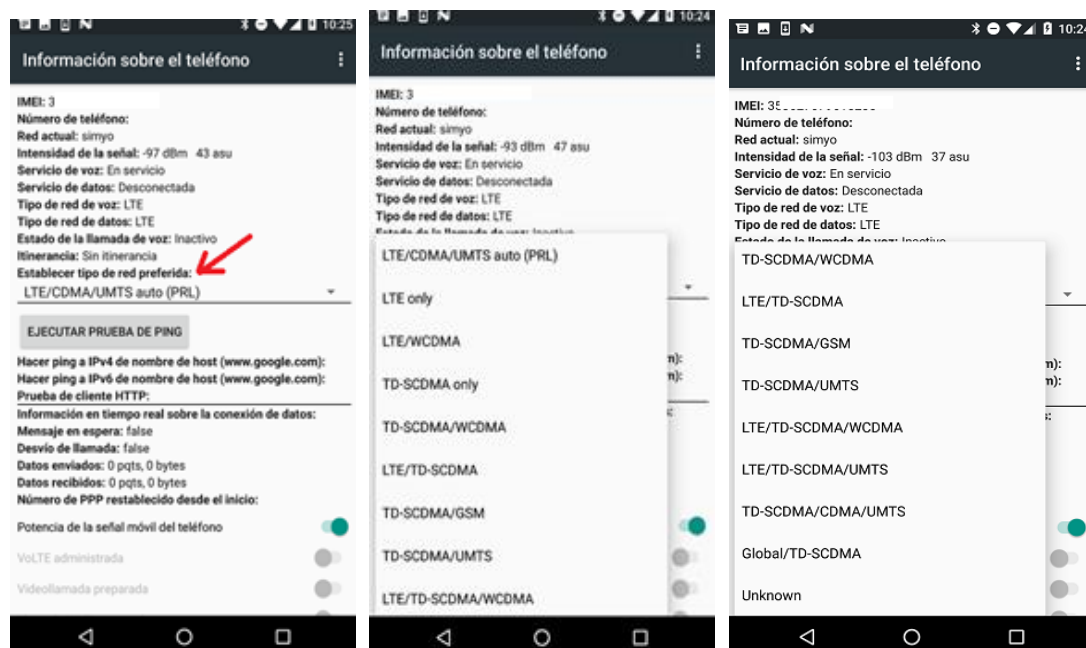
18.4. SELECCIÓN DE LA RED DE DATOS MÓVILES

1799. Las redes de telefonía 2G (o GSM) presentan diferentes vulnerabilidades en la actualidad, como por ejemplo la posibilidad de suplantación de la red por parte de un potencial atacante debido a debilidades en los mecanismos de autenticación, o la posibilidad de capturar y descifrar el tráfico de voz GSM debido a debilidades en los algoritmos de cifrado, A5/1 [Ref.- 400].
1800. Con el objetivo de mitigar las vulnerabilidades existentes actualmente en las redes de telefonía móvil 2G (o GSM), se recomienda forzar al dispositivo móvil a conectarse únicamente, tanto para comunicaciones de voz como de datos, a las redes de telefonía 3G (UMTS) o 4G).
1801. Android no permite restringir las comunicaciones de telefonía móvil sólo a redes 3G o 4G a través del interfaz de usuario estándar (ver opciones mediante el menú de prueba o *testing* posteriormente).
1802. A través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - Redes móviles", y en concreto de la opción "Tipo de red preferida", es posible indicar que sólo se haga uso de redes 2G (GSM - GPRS o EDGE), una opción indicada para disminuir el consumo de batería del terminal, pero desaconsejada desde el punto de vista de seguridad:



1803. La opción por defecto es "4G (recomendado)". Se aconseja no modificar esta opción, especialmente al valor "2G", pese a que el dispositivo móvil hará uso automáticamente de redes 2G en zonas donde no se disponga de suficiente cobertura por parte de las redes 3G o 4G.
1804. El principal inconveniente de esta configuración (no modificable) es que si el usuario se encuentra en una zona en la que no se dispone de cobertura 3G (o superior, 3.5G o 4G), se podrá hacer uso del dispositivo móvil a través de 2G, premiando la disponibilidad del servicio de telefonía (voz y datos) por encima de la seguridad.

1805. Android proporciona información en la barra superior de estado a través de un conjunto de iconos específico sobre las tecnologías de telefonía móvil de datos para las que se dispone de cobertura, que representan la disponibilidad y el estado de la conexión para una red 4G, 3G o 3.5G (WCDMA/UMTS o HSPA), 2.75G (EDGE) y 2G (GPRS).
1806. La cobertura y conectividad existentes están identificadas por un icono con forma de triángulo con el vértice hacia la izquierda (formado por barras en progresión ascendente), que representa la cobertura existente en función del número de barras activas, o que no se dispone de cobertura si el icono con forma de triángulo está vacío (no dispone de ninguna barra).
1807. La configuración por defecto del dispositivo móvil premia la conectividad, permitiendo el uso de redes de telefonía móvil 2G, 3G o 4G de forma automática en función de su cobertura y disponibilidad.
1808. Adicionalmente, es posible obtener y configurar información avanzada a través del menú de prueba (o *testing*), disponible en Android tras marcar el siguiente número de teléfono: `*##*#4636*##` (o `*##*#info*##`). Se debe ser muy cuidadoso a la hora de hacer uso de las diferentes opciones que este menú ofrece, ya que permite cambiar opciones asociadas a la telefonía que podrían afectar a la funcionalidad del terminal, como por ejemplo la selección de bandas de frecuencia de USA, no válidas en Europa.
1809. Mediante la selección de la opción "Información sobre el teléfono" se dispone de numerosos detalles de la red de telefonía móvil, siendo posible configurar el tipo de red preferida mediante la opción "Establecer tipo de red preferida":



1810. De las diferentes opciones disponibles, desde el punto de vista de seguridad se recomienda seleccionar la opción "LTE only" o "WCDMA only" para forzar al dispositivo móvil a conectarse únicamente a redes 4G o 3G. Por defecto la opción seleccionada es "LTE/CDMA/UMTS auto (PRL)", donde

el término PRL hace referencia a la "Preferred Roaming List" definida por los operadores de telefonía móvil.

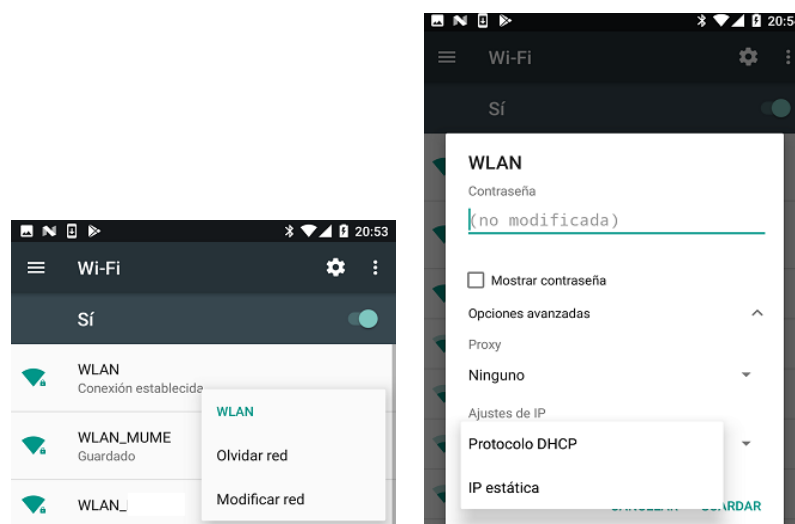
1811. Este cambio de configuración prevalece incluso tras reiniciar el dispositivo móvil, y fuerza a que el dispositivo móvil sólo haga uso de redes 4G o 3G, más seguras que las redes 2G.
1812. Se debe ser muy cauteloso a la hora de acceder al menú de prueba, y únicamente aplicar las recomendaciones expuestas en la presente guía, dado que algunas de ellas pueden provocar comportamientos inesperados en el dispositivo móvil. En cualquier caso, si se opta por realizar cambios sobre esta configuración, se deberá apuntar cuáles eran los valores originales para poder revertir a ellos en caso de comportamientos inestables de los servicios de voz y datos.

19. COMUNICACIONES TCP/IP

1813. Esta sección proporciona recomendaciones de seguridad para la pila de comunicaciones TCP/IP del dispositivo móvil, incluyendo recomendaciones generales para TCP/IP (IPv4), SSL/TLS, IPv6 y redes VPN.

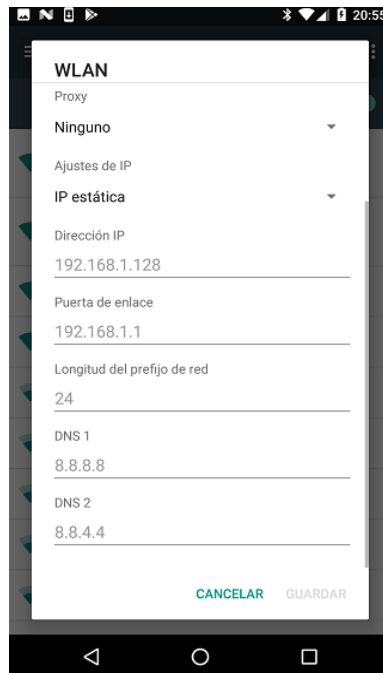
19.1. ADAPTADORES DE RED

1814. La información del adaptador de red disponible en el dispositivo móvil para las comunicaciones mediante tecnologías Wi-Fi (interfaz Wi-Fi) se puede consultar a través del menú "Ajustes [Conexiones Inalámbricas y Redes] - Wi-Fi - [Doble pulsación sobre el nombre de la red] - Modificar red". Se requiere haber establecido previamente una conexión con dicha red Wi-Fi, aunque no sea la conexión activa actualmente.

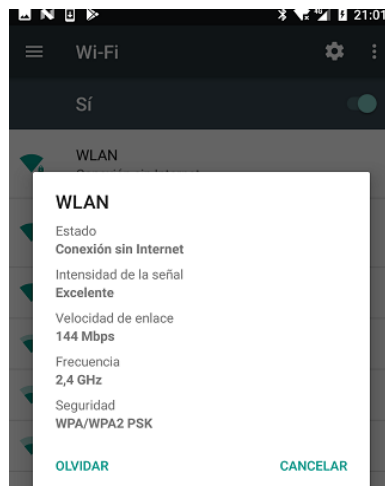


1815. La pantalla de ajustes avanzados permite seleccionar si se utilizará una dirección IP estática o dinámica (DHCP). Por defecto, la configuración de TCP/IP empleada para la conexión a las redes Wi-Fi hace uso de una dirección IP dinámica obtenida mediante el protocolo DHCP.

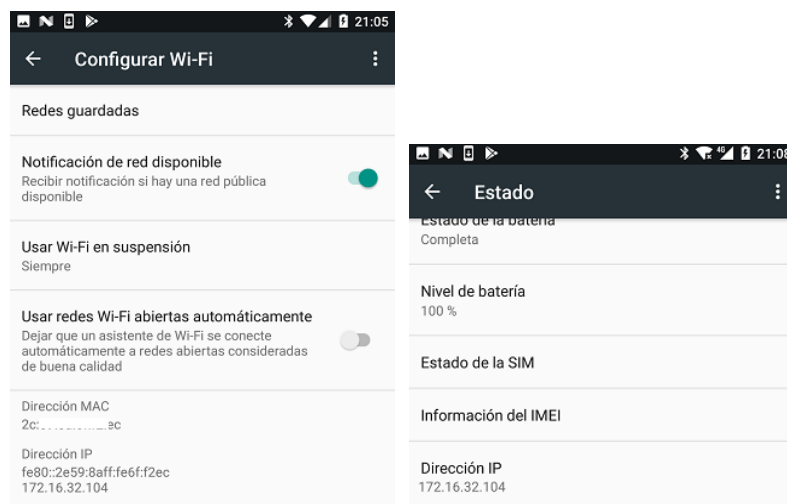
1816. La configuración de la dirección IP estática permite definir la dirección IP, la puerta de enlace (o router por defecto), la máscara (o prefijo) de red, así como los (dos) servidores DNS:



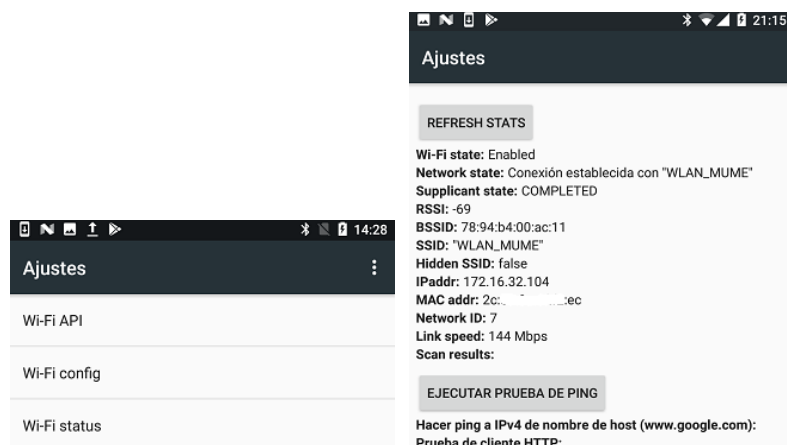
1817. La configuración del direccionamiento IP del interfaz Wi-Fi es independiente para cada una de las redes Wi-Fi configuradas (desde la versión 4.0 de Android, "Ice Cream Sandwich"), en lugar de disponerse de una configuración o perfil de red único para todas las red Wi-Fi, tal como ocurría en versiones previas de Android, como 2.x [Ref.- 96].
1818. En el caso de emplear una dirección IP asignada dinámicamente por DHCP, es posible obtener información de la misma a través de la entrada correspondiente a la red inalámbrica con la que se ha establecido la conexión Wi-Fi. Desde el menú "Ajustes [Conexiones Inalámbricas y Redes] - Wi-Fi, se muestra la lista de redes Wi-Fi visibles y aquella a la que estamos conectados.
1819. Una pulsación breve sobre el nombre de la red mostrará una pantalla con los datos de conexión a dicha red, pero, al igual que sucedía en Android 6.x, Android 7.x tampoco muestra la dirección IP que ha sido asignada al dispositivo móvil por el servidor DHCP, a diferencia de lo que ocurría en versiones previas de Android hasta la 4.4.x.



1820. Para acceder a la información de la dirección IP es necesario acceder al menú "Ajustes - [Conexiones inalámbricas y redes] Wi-Fi [rueda de engranaje]", o consultar la información de estado del dispositivo móvil, a través de "Ajustes - [Sistema] Información del teléfono - Estado":



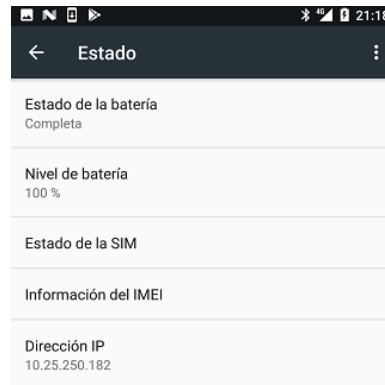
1821. Adicionalmente, es posible obtener esta información a través del menú de prueba (o testing), disponible en Android tras marcar el siguiente número de teléfono: `*##4636##` (o `*##info##`):



1822. La opción "Wi-Fi information - Wi-Fi status", tras pulsar el botón "Refresh stats" en caso de no disponer de datos, proporciona los detalles de

conexión de la red Wi-Fi actual, en el ejemplo, red "WLAN_MUME" (ver imágenes superiores).

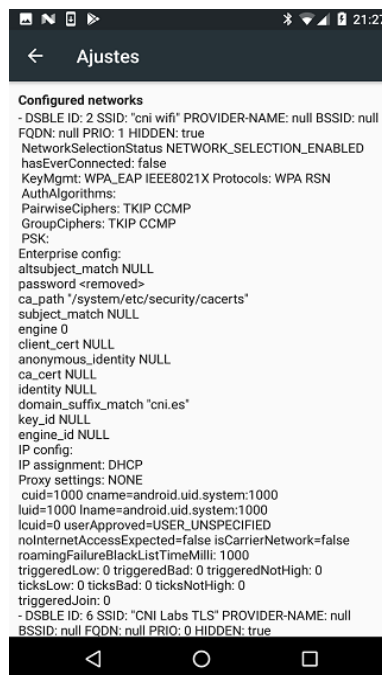
1823. En caso de conexión a redes de datos de telefonía móvil 2/3/4G, la dirección IP asignada por el operador se puede comprobar a través de "Ajustes - [Sistema] Información del teléfono - [Estado] Dirección IP":



1824. También existe la posibilidad de emplear el comando "ifconfig" (previa invocación del comando "adb shell" desde un ordenador). El direccionamiento IP se muestra como "wlan0" para el interfaz Wi-Fi ("wlan0") y como "rmnet_data0" para el interfaz de telefonía:

```
rmnet_data0 Link encap:UNSPEC
            inet addr:10.25.250.182  Mask:255.255.255.252
            inet6 addr: fe80::1c0b:f0f0:ee6f:2f61/64 Scope: Link
            UP RUNNING MTU:1340 Metric:1
            RX packets:428 errors:0 dropped:0 overruns:0 frame:0
            TX packets:546 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:96984 TX bytes:84880
```

1825. La opción "Wi-Fi information - Wi-Fi config" proporciona los detalles de todas las redes que forman parte de la PNL (en la que se incluyen las redes a las que el dispositivo móvil se ha conectado y las que se han añadido de forma manual). Entre los detalles que se muestran, se incluye el tipo de seguridad Wi-Fi de la red y, si procede, la relación de certificados digitales asociados a cada red:



19.2. SEGURIDAD EN TCP/IP

1826. Con el objetivo de incrementar el nivel de seguridad de la pila TCP/IP empleada para las conexiones de datos en Android, se recomienda evaluar su comportamiento y (de ser posible) modificar ciertos parámetros de configuración que condicionan su funcionamiento y nivel de seguridad.
1827. Desde el punto de vista de seguridad, se recomienda deshabilitar el procesamiento de paquetes de *ping*, es decir, paquetes ICMP *echo request* entrantes y sus respuestas asociadas, paquetes ICMP *echo response* salientes, con el objetivo de limitar el descubrimiento del dispositivo móvil en la red (habilitados por defecto en Android).
1828. Sin embargo, no se ha identificado ningún mecanismo ni opción de configuración en Android que permita deshabilitar la recepción de paquetes ICMP *echo request* y sus respuestas ICMP *echo response*.
1829. Tampoco se han encontrado mecanismos que permitan modificar los ajustes de configuración de bajo nivel de la pila TCP/IP mencionados en este apartado, por lo que el mismo se proporciona como referencia para el usuario y de cara a tenerlo en cuenta por si futuras versiones ofreciesen su parametrización.
1830. Android también responde a otros tipos de mensajes ICMP de petición y respuesta, como por ejemplo ICMP *timestamp*, aunque no responde a mensajes ICMP *address subnet mask*.
1831. La identificación del dispositivo móvil también se puede llevar a cabo a través del protocolo DHCP, ya que en las peticiones DHCP (DHCP *Discover* y *Request*) Android incluye varios datos identificativos.
1832. Las peticiones de Android 7.1.2 añaden la opción 60 de DHCP, identificación del vendedor (*Vendor class identifier*), con el mensaje

"android-dhcp-7.1.2", es decir, revelan que el cliente DHCP es un dispositivo Android y su versión.

1833. Debe tenerse en cuenta que la última versión disponible de `dhcpcd` para Linux puede ser mucho mayor a la utilizada por Android, y solucionar diferentes vulnerabilidades de seguridad conocidas públicamente [Ref.- 47].

Nota: En versiones anteriores de Android, se utilizaba el demonio estándar de Linux DHCPD (DHCP Client Daemon) versión 5.5.6. En septiembre de 2014 se publicó una vulnerabilidad de denegación de servicio (DoS) en `dhcpcd` que afectaba a todas las versiones desde la 4.0.0 hasta la 6.4.2. Se confirmó que la versión 5.5.6 de `dhcpcd` disponible en Android 4.4.x era vulnerable, estando solucionada esta vulnerabilidad en Android 5.0 (Lollipop), pese a hacer uso del mismo número de versión de `dhcpcd` [Ref.- 196]. Más información:

<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6060>

<http://roy.marples.name/projects/dhcpcd/ci/1d2b93aa5ce25a8a710082fe2d36a6bf7f5794d5?sbs=0>

1834. Las peticiones también incluyen la opción 12 de DHCP, nombre de equipo (*Host Name*), con el valor "android_<código>", donde "<código>" es un string de 16 caracteres hexadecimales correspondiente al `ANDROID_ID` (asociado a la propiedad "net.hostname"; ver apartado "7.12. Configuración por defecto del dispositivo móvil").

1835. Adicionalmente, por defecto el dispositivo móvil Android envía peticiones NTP (*Network Time Protocol, UDP/10203*) para su sincronización horaria hacia servidores NTP disponibles en Internet periódicamente.

19.3. ESCANEOS DE PUERTOS TCP Y UDP

1836. A continuación, se analiza el comportamiento de la pila TCP/IP de Android frente a intentos de conexión TCP y UDP a través del interfaz Wi-Fi. El análisis se ha realizado en la configuración por defecto del dispositivo móvil, es decir, sin haber realizado ninguna configuración específica o instalación adicional de software.
1837. Android responde con un paquete TCP *Reset* a las peticiones de conexión hacia puertos TCP cerrados.
1838. Por defecto, Android no dispone de ningún puerto TCP abierto o filtrado, es decir, los 65.536 puertos TCP están cerrados.
1839. Android responde con paquetes ICMP de tipo 3 y código 3 (destino inalcanzable, puerto inalcanzable), limitados a uno por segundo, a las peticiones de conexión hacia puertos UDP cerrados.
1840. Por defecto, Android solo tiene el puerto UDP/5353 (correspondiente a multicast DNS, mDNS o Bonjour) abierto o filtrado, tanto si DHCP está configurado en el interfaz Wi-Fi como cuando se utiliza una dirección IP estática. El resto de puertos UDP están cerrados.
1841. El comportamiento por defecto de la pila TCP/IP, y especialmente la ausencia de mecanismos de filtrado de tráfico (cortafuegos personal) y sus

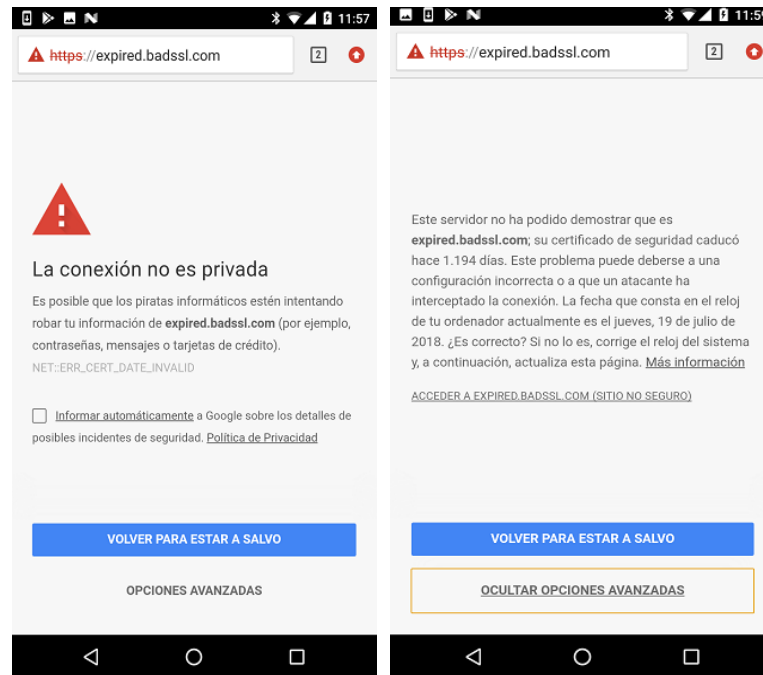
peculiaridades a nivel de TCP y UDP, podrían hacer posible la identificación de los dispositivos móviles basados en Android.

1842. Este tipo de identificación remota a través de la red facilitaría a un potencial atacante la realización de ataques dirigidos sobre vulnerabilidades conocidas, por lo que se enfatiza la necesidad de mantener el software del dispositivo móvil completamente actualizado en todo momento.

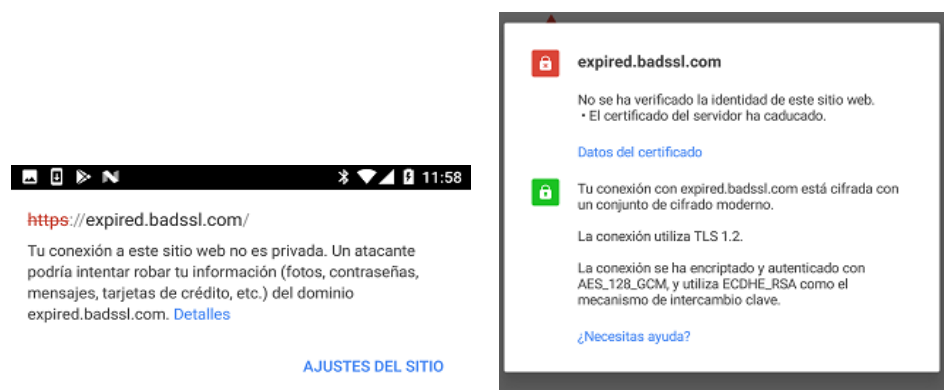
Nota: El comportamiento de la pila TCP/IP frente a escaneos de puertos a través del interfaz de telefonía móvil de datos (cuando está conectado a la red real del operador) no puede ser analizado de forma fiable. Este comportamiento a través del interfaz de telefonía móvil de datos está condicionado por la infraestructura de comunicaciones y los mecanismos de filtrado de los operadores de telefonía móvil, que por ejemplo, pueden no permitir comunicaciones mediante ping (*ICMP echo request y reply*) o pueden disponer de *proxies* transparentes en la red, por ejemplo para protocolos como FTP o HTTP.

19.4. HTTPS: SSL/TLS

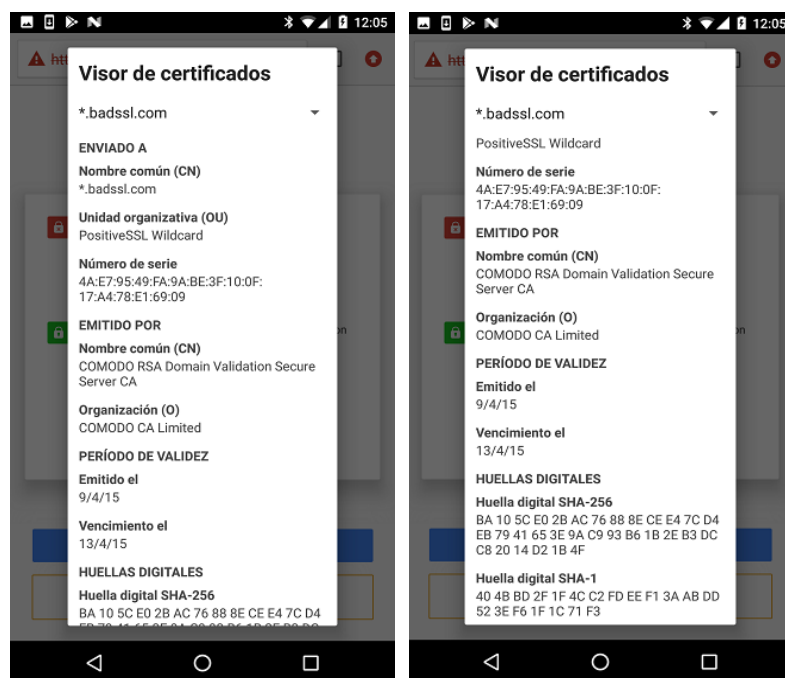
1843. La implementación de los protocolos SSL, Secure Socket Layer, y TLS, Transport Layer Security, en Android combina capas implementadas en C (BoringSSL, derivado de OpenSSL) y en Java (Conscrypt, OkHttp, y Java Secure Sockets, entre otras librerías), con soporte para SSL v3.0 (desde la API 10 hasta la 25) y hasta TLS v1.2 desde la API 16 en adelante [Ref.- 102].
1844. Desde Android 6.x, Google migró internamente a una implementación propia de la librería de SSL y TLS denominada BoringSSL, empleada también por Chromium y otros servicios de Google [Ref.- 605].
1845. Desde enero de 2017, al acceder a un sitio web mediante HTTPS utilizando el navegador web existente en Android por defecto (app "Chrome" en Android 6.x), el dispositivo móvil intentará verificar el certificado digital asociado y su cadena de confianza, hasta la autoridad certificadora (CA, *Certificate Authority*) raíz reconocida correspondiente.
1846. En caso de que no sea posible verificar la validez de un certificado, el navegador web de Android generará un mensaje de advertencia, indicando el motivo del error y permitiendo al usuario ver y analizar algunos detalles del motivo a través del enlace de opciones avanzadas:



1847. Sin embargo, la información detallada del certificado digital obtenido no está disponible directamente. Es necesario pulsar el icono superior de color rojo con un triángulo, disponible en la barra de navegación o URL, y seleccionar "Detalles" para así poder obtener la información del certificado digital:



1848. El enlace "Datos del certificado" permite obtener los detalles del certificado digital a través del "Visor de certificados", que incluye elementos clave para su verificación, como por ejemplo el número de serie del certificado, las huellas o *fingerprints* SHA-1 o SHA-256, información de la autoridad certificadora involucrada en su emisión, etc.:



1849. La pantalla con el enlace "Datos del certificado" también puede proporcionar, en su parte inferior, algunos detalles de la conexión TLS y de los algoritmos y parámetros negociados entre el navegador web y el servidor web (ver imágenes superiores).
1850. Si se acepta un certificado digital temporalmente, éste será válido hasta que se cierre la instancia, proceso, o sesión actual del navegador web en Android, ya que Chrome mantiene una caché en memoria con éste.
1851. El aceptar un certificado que no es de confianza tiene el riesgo asociado de que el certificado digital aceptado, y desconocido, puede pertenecer a un potencial intruso que se encuentra realizando un ataque de interceptación del tráfico del usuario, MitM (Man-in-the-Middle), sobre el protocolo SSL/TLS y las comunicaciones (supuestamente) cifradas.
1852. Como se describe en el apartado "11. Gestión de certificados digitales y credenciales", el repositorio de certificados raíz en Android 7.x está accesible a través del interfaz de usuario de Android. Por tanto, el usuario puede gestionar la lista autoridades certificadoras (o CAs, *Certificate Authorities*) que son consideradas de confianza por Android.
1853. Android desde la versión 4.0 (Ice Cream Sandwich) soporta la utilización de mecanismos de autenticación web basados en certificados digitales cliente a través del navegador web existente por defecto [Ref.- 120], al igual que es posible hacer uso de certificados cliente para la conexión a redes Wi-Fi o redes VPN (ver apartado "11. Gestión de certificados digitales y credenciales").
1854. La funcionalidad para el uso de certificados digitales cliente en el navegador web añadida finalmente en Android versión 4.0 [Ref.- 121] permite gestionar (visualizar, deshabilitar, habilitar, instalar, eliminar, etc.) las CAs desde el interfaz de usuario de Android y la sección "Ajustes - Seguridad", así como gestionar certificados digitales cliente para el

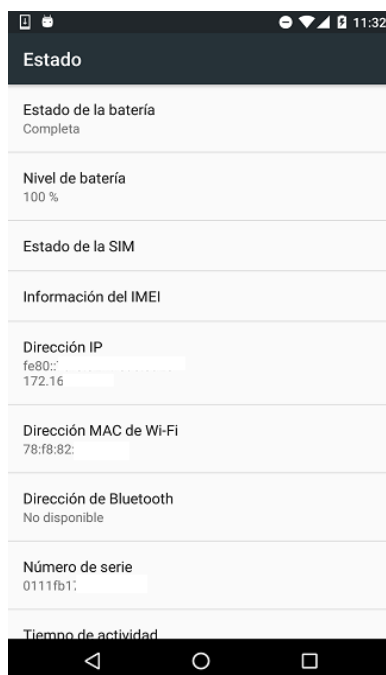
dispositivo móvil y las apps (como el navegador web o los clientes de correo electrónico) a través de un nuevo KeyChain (o gestor de credenciales y certificados) [Ref.- 122] [Ref.- 188].

1855. Previamente se disponía de un gestor de credenciales y certificados global para todo el sistema pero únicamente accesible para la conexión a redes Wi-Fi y VPN, a través de "Ajustes" [Ref.- 188]. Las apps debían hacer uso de sus propios gestores de certificados privados, ya que no disponían de una API para acceder a los certificados existentes en el repositorio del sistema.
1856. En Android 4.x, los certificados digitales cliente guardados en el repositorio de credenciales están disponibles para su utilización por parte del navegador web [Ref.- 185].
1857. Existen otros navegadores web de terceros (como por ejemplo Firefox para Android, junto al *add-on* o extensión Cert Manager¹¹⁶, SandroB, etc.), que también disponen de capacidades de autenticación web basada en certificados digitales cliente, y que adicionalmente definen su propio gestor de certificados cliente (KeyManager) y su propio gestor de confianza (TrustManager, para gestionar sus propias CAs) a través de "javax.net.ssl.SSLContext".
1858. Android 5.x introdujo cambios en la configuración de TLS/SSL empleada, por ejemplo, para las conexiones HTTPS (entre otras), incluyendo soporte para TLS v1.1 y v1.2, AES-GCM (AEAD), y dando preferencia a los algoritmos con capacidades de *forward secrecy*, como ECDHE y DHE. Por otro lado, se deshabilita el soporte para MD5, 3DES, y los algoritmos ECDH con claves estáticas.
1859. Desde Android 6.x, el AKS (*Android Keystore System*) mejora la gestión del acceso al material criptográfico del dispositivo móvil, permitiendo a las apps acceder a los certificados digitales del sistema, de usuario (para uso en redes VPN y Wi-Fi), de cliente o personales (por ejemplo, para el navegador web, el cliente de correo, etc.), e incluso a claves secretas con cifrado simétrico (ver apartado "11.1. Android Keystore System: Almacenamiento de credenciales").
1860. El proveedor de seguridad de Google Play Services también ofrece estas capacidades para versiones de Android previas hasta la 2.3.
1861. El uso de SSLSocketFactory y SSLSocket directamente por parte de una app puede emplearse para establecer comunicaciones cifradas de bajo nivel sin hacer uso de la configuración y prerequisites del sistema previamente indicados, permitiendo el uso de algoritmos más inseguros.

¹¹⁶ <https://addons.mozilla.org/en-US/mobile/addon/cert-manager/>

19.5. IPV6

1862. El protocolo IP versión 6 (IPv6) está soportado y activado por defecto en Android para las comunicaciones de datos mediante Wi-Fi y telefonía móvil 2/3/4G.
1863. En Android 7.x el usuario dispone de capacidades para visualizar la información de las direcciones IPv6 a través del menú "Ajustes - Información del teléfono - Estado", y en concreto, mediante el campo "Dirección IP"¹¹⁷, aunque no dispone de ninguna opción de configuración relacionada con IPv6 desde el interfaz gráfico de usuario (o GUI):



Nota: La disponibilidad de IPv6 para las comunicaciones de datos a través de la infraestructura de telefonía móvil en Android es dependiente tanto de la versión del sistema operativo, como del modelo de dispositivo móvil y del operador de telefonía móvil empleados [Ref.- 53].

1864. Por otro lado, y desde el punto de vista de la funcionalidad, es necesario evaluar si Android soporta únicamente redes de datos basadas en IPv6, sin hacer uso de la pila TCP/IP versión 4 [Ref.- 56]. Las redes de telefonía móvil 2/3/4G basadas sólo en IPv6 están soportadas por Android, y las redes Wi-Fi también lo están, aunque inicialmente con problemas conocidos de conectividad relacionados con la coexistencia de IPv6 e IPv4.
1865. Para disponer de conectividad IPv6 completa en Android y obtener una dirección IPv6 a través de mensajes RA (Router Advertisement) enviados por un router con soporte de IPv6 es necesario que el interfaz Wi-Fi disponga también de dirección IPv4 (estática o mediante DHCP).
1866. Es posible forzar el uso exclusivo de IPv6 en el interfaz Wi-Fi de Android, y evitar los problemas conocidos mencionados, estableciendo una dirección

¹¹⁷ <https://plus.google.com/+KevinOtte/posts/JUHVxyW1AB3>

IP fija para la red Wi-Fi igual a 0.0.0.0, el mismo valor que también debe ser asignado al *gateway* [Ref.- 56].

Nota: Las limitaciones del uso exclusivo de IPv6 en el interfaz Wi-Fi de Android están relacionadas también con problemas en el soporte para otras funcionalidades de IPv6:

- RFC3315 (DHCPv6): <https://code.google.com/p/android/issues/detail?id=32621>
- RFC6106 (RDNSS): <https://code.google.com/p/android/issues/detail?id=32629> ¹¹⁸

1867. Durante el proceso de inicialización de conectividad de datos, el dispositivo móvil Android genera peticiones IPv6 de solicitud de vecinos y de router mediante ICMPv6, empleando como dirección origen "fe80::6489:9aff:fe01:0203" (donde su dirección MAC es 64:89:9a:01:02:03), y añadiendo también su dirección MAC como dirección de enlace.

1868. La utilización de la dirección MAC en la dirección IPv6 sin hacer uso de las Privacy Extensions de IPv6 (RFC 4941) presenta problemas de seguridad y permite el seguimiento de los dispositivos móviles Android en Internet [Ref.- 54].

1869. Finalmente, Android 4.0 añadió soporte para las Privacy Extensions de IPv6, pero sin capacidades de configuración, ya que su utilización está configurada por defecto, pero no puede ser deshabilitada en caso de que sea necesario (aunque no recomendado desde el punto de vista de seguridad) [Ref.- 58].

1870. Adicionalmente, el dispositivo móvil Android envía periódicamente mensajes IPv6 de informes multicast ("Multicast Listener Report Message v2").

1871. Se recomienda deshabilitar la disponibilidad de IPv6 en Android por completo, salvo que se vaya hacer uso de sus capacidades de comunicación, aunque desafortunadamente esta opción no está disponible actualmente en la versión de Android analizada (salvo en dispositivos Android *rooteados*).

Nota: No se ha identificado ningún mecanismo, ni opción de configuración, en Android que permita deshabilitar o gestionar la pila de comunicaciones de IPv6.

19.6. REDES VPN

1872. Android, desde la versión 4.0 (Ice Cream Sandwich), proporciona soporte para las siguientes tecnologías y protocolos empleados por redes VPN (*Virtual Private Networks*, o redes privadas virtuales) [Ref.- 21]:

- PPTP: Point-to-Point Tunneling Protocol (TCP/1723).
- L2TP/IPSec: Layer Two Tunneling Protocol (L2TP; UDP/1701) en combinación con Internet Protocol Security (IPSec; protocolos IP 50 y 51 + IKE: UDP/500).
 - L2TP/IPSec PSK
 - L2TP/IPSec RSA
- IPSec: Internet Protocol Security (protocolos IP 50 y 51 + IKE: UDP/500).

¹¹⁸ Resuelto en Android 5.0 (Lollipop): <https://code.google.com/p/android/issues/detail?id=32630>

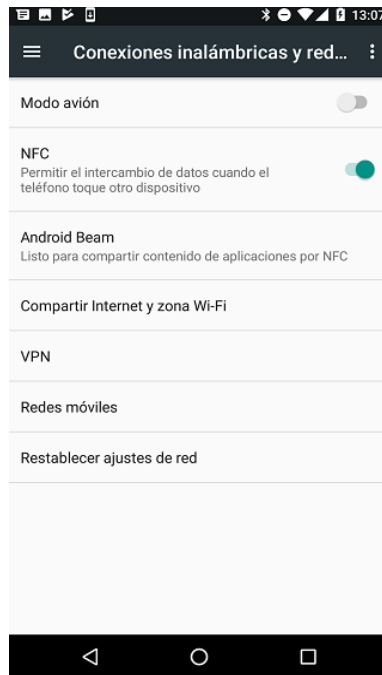
- IPSec Xauth PSK
 - IPSec Xauth RSA¹¹⁹
 - IPSec Hybrid RSA¹²⁰
1873. La opción recomendada desde el punto de vista de seguridad es hacer uso de redes VPN basadas en IPSec o L2TP/IPSec.
1874. Desde el punto de vista de seguridad no se recomienda hacer uso de redes VPN basadas en PPTP, ya que este protocolo hace uso de MSCHAPv2 para el proceso de autenticación, y éste es vulnerable a ataques de fuerza bruta con un 100% de éxito (representado por la totalidad de posibilidades asociadas a una clave DES, es decir, 2^{56})¹²¹.
1875. Adicionalmente, en el caso hacer uso de redes VPN basadas en L2TP/IPSec o IPSec, se debe definir el método de autenticación a emplear: certificados digitales (opción "RSA") o clave precompartida (opción "PSK", *Pre-Shared Key*).
1876. Se recomienda emplear certificados para la autenticación de L2TP/IPSec o IPSec al ser un mecanismo más seguro que el establecimiento de una clave precompartida (PSK) entre el servidor de VPN y el dispositivo móvil (que además es conocida por todos los usuarios de la red VPN y puede ser usada en ataques MitM¹²²).
1877. En el caso de IPSec Xauth RSA tanto el cliente como el servidor se autentican inicialmente empleando certificados digitales durante la fase 1 de IKE (autenticación mutua). Posteriormente, el cliente se autentifica de nuevo mediante Xauth (ej. usuario y contraseña). En el caso de IPSec Hybrid RSA el cliente no se autentifica durante la fase 1 de IKE, sino que sólo verifica que está hablando con el servidor legítimo, lo que simplifica el despliegue de la solución VPN al no ser necesaria la emisión de certificados digitales cliente, pero constituye una solución más robusta que IPSec Xauth PSK. Posteriormente, de nuevo, el cliente se autentifica mediante Xauth (ej. usuario y contraseña).
1878. Desde el menú "Ajustes [Conexiones Inalámbricas y Redes] - Más... - VPN" es posible añadir una nueva red VPN a través del botón con el símbolo "+" de la parte superior:

¹¹⁹ <http://tools.ietf.org/html/draft-ietf-ipsec-isakmp-xauth>

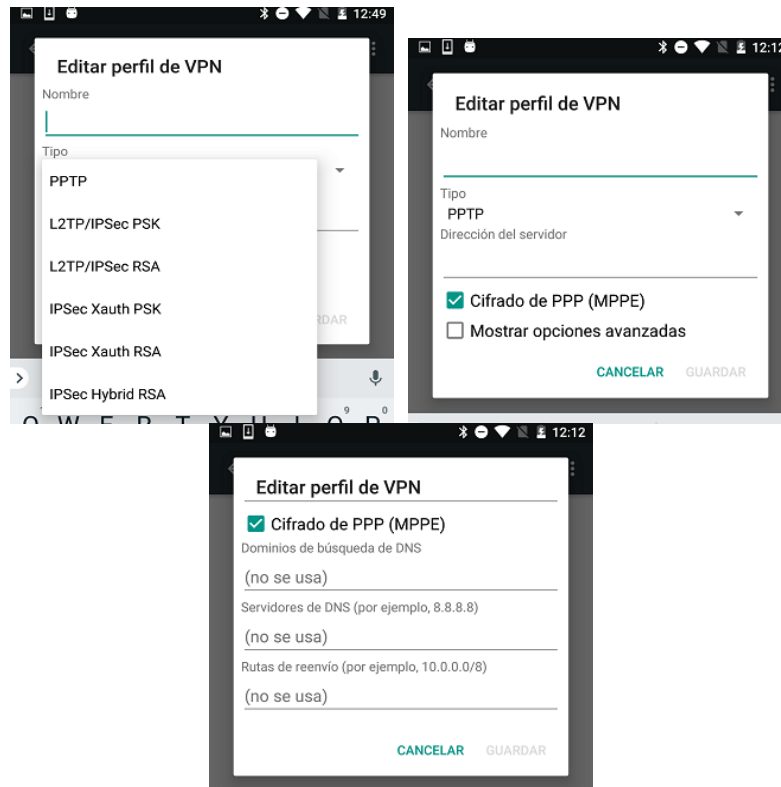
¹²⁰ <http://tools.ietf.org/html/draft-ietf-ipsec-isakmp-hybrid-auth>

¹²¹ <https://www.cloudcracker.com/dictionaries.html> (*servicio no disponible actualmente*).

¹²² http://www.cisco.com/en/US/tech/tk583/tk372/technologies_security_notice09186a0080215981.html

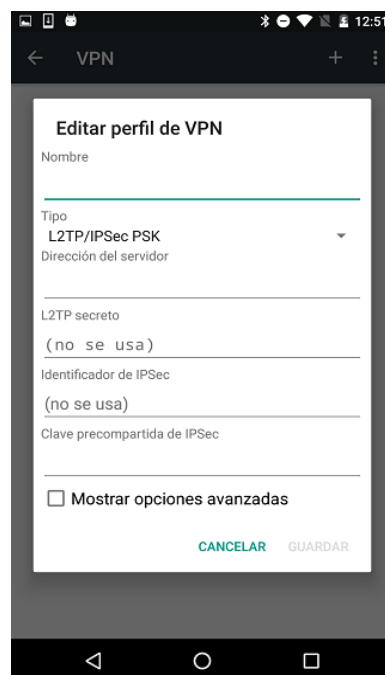


1879. A través de las diferentes opciones disponibles asociadas al perfil de la red VPN es posible configurar todos los detalles de la conexión VPN. Una vez se han establecido las opciones de configuración según el tipo de red (descritas a continuación), mediante el botón "Guardar" es posible almacenar la nueva configuración.
1880. Todas las redes VPN, basadas en PPTP, L2TP/IPSec e IPSec, requieren disponer del nombre de la red VPN, del tipo de red VPN y del nombre o dirección IP del servidor VPN.
1881. En el caso de redes PPTP, Android permite deshabilitar el cifrado mediante la opción "Encriptación de PPP (MPPE)", que está habilitada por defecto. Se desaconseja deshabilitar este mecanismo de cifrado desde el punto de vista de seguridad:



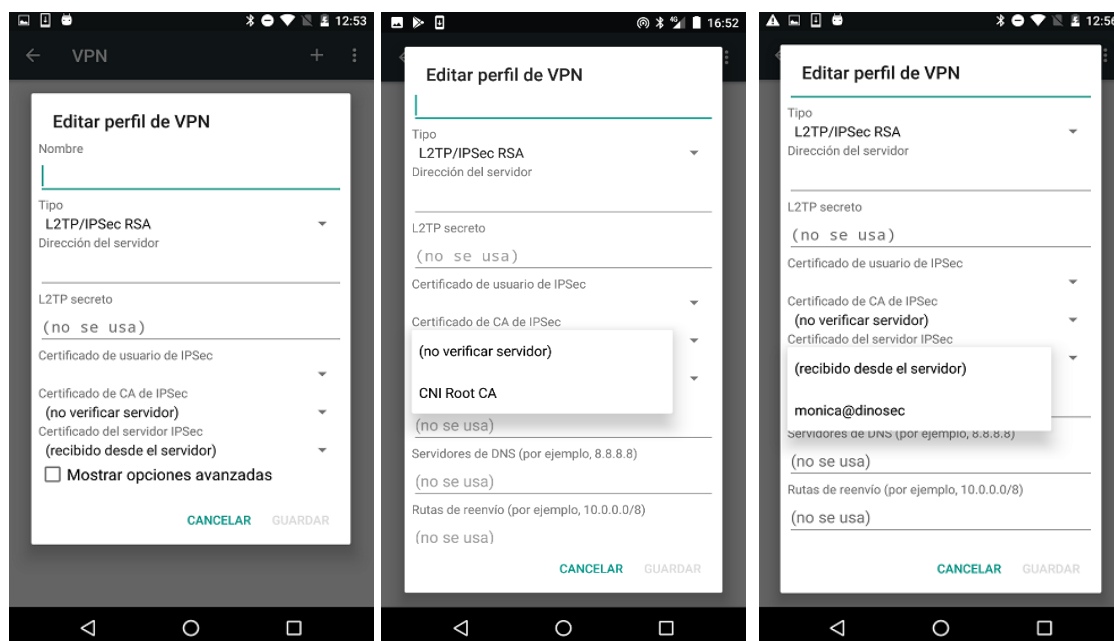
1882. La opción "Mostrar opciones avanzadas" permite especificar los dominios de búsqueda de DNS, los servidores DNS, y posibles rutas para el envío del tráfico de red, es decir, modificar elementos relacionados con la pila TCP/IP de Android y, en concreto, con la resolución de nombres (DNS) y la tabla de enrutamiento de Android, como se ilustra en la imagen superior derecha.

1883. En el caso de redes L2TP/IPSec PSK, y de redes IPsec Xauth PSK, Android permite establecer la clave precompartida para el acceso a la red:



1884. Tanto para redes L2TP/IPSec PSK como RSA, Android, en función del tipo de red VPN, también permite habilitar un secreto L2TP (deshabilitado por defecto), opción que facilita autenticar el propio túnel L2TP mediante un secreto compartido entre el cliente y el servidor (y que puede ser necesario en función de la configuración del servidor VPN).

1885. En el caso de redes L2TP/IPSec RSA, y de redes IPSec Xauth RSA, Android permite establecer el certificado digital cliente de usuario de IPSec, el certificado de la autoridad certificadora (CA) que ha emitido el certificado del servidor o concentrador VPN, y específicamente, el certificado digital del servidor o concentrador VPN (si no se especifica un certificado digital concreto, Android por defecto aceptará el certificado recibido al establecer la conexión, opción desaconsejada desde el punto de vista de seguridad):



1886. Por defecto, Android no obliga a que la configuración de la red VPN basada en L2TP/IPSec RSA verifique el certificado de la CA que ha emitido el certificado del servidor o concentrador VPN.

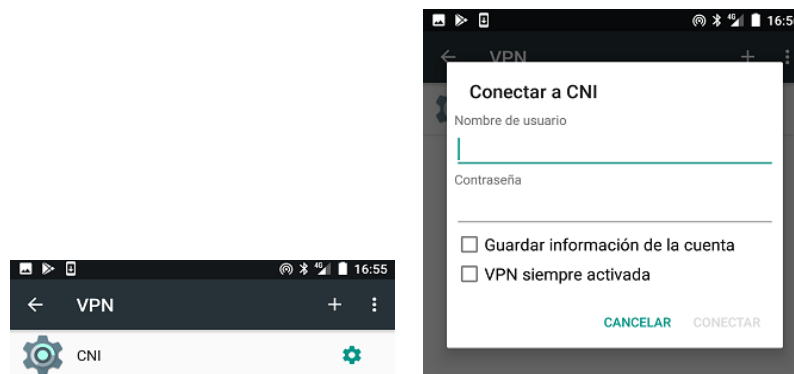
1887. Esta es la opción desaconsejada desde el punto de vista de seguridad ya que, al no verificarse la CA, se permite que un potencial atacante suplante al servidor o concentrador VPN y se haga pasar por el servidor legítimo. En su lugar, siempre se debe seleccionar el certificado digital de la CA que está siendo empleado por la red VPN, y que debe haber sido importado previamente en el almacén de certificados de Android (ver apartado "11. Gestión de certificados digitales y credenciales").

1888. El certificado digital cliente necesario para L2TP/IPSec es de tipo personal, y permitirá autenticar al dispositivo móvil y a su propietario. Adicionalmente, como se ha recomendado previamente, es necesario instalar un certificado raíz (perteneciente a la CA), y asociado a la generación del certificado digital empleado por el servidor de la red VPN.

1889. En el caso de redes L2TP/IPSec basadas en certificados digitales, al seleccionar el certificado cliente del usuario y de la CA, Android mostrará los

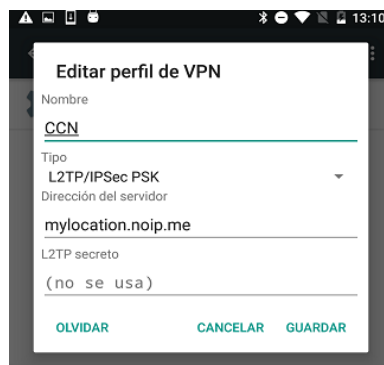
certificados disponibles en el repositorio de credenciales, previamente importados (ver apartado "11. Gestión de certificados digitales y credenciales").

1890. En el caso de redes IPsec Hybrid RSA se pueden definir los certificados de la CA y del servidor o concentrador VPN, pero no se hace uso de un certificado digital cliente ni de una clave precompartida (opciones Xauth previas).
1891. Una vez configurada la red VPN, es posible conectarse a la misma seleccionándola de la lista de redes disponibles desde la pantalla "Ajustes [Conexiones Inalámbricas y Redes] - Más... - VPN", mediante una pulsación corta sobre la misma:

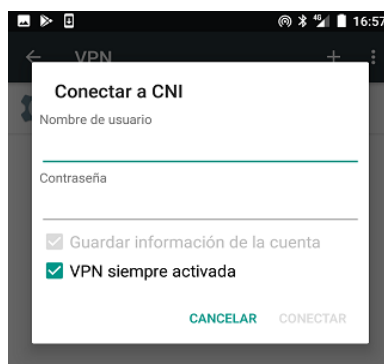


1892. Como resultado, Android solicitará al usuario las credenciales de acceso a la red VPN, es decir, el nombre de usuario y la contraseña para establecer la conexión con la red VPN seleccionada.
1893. Desde el punto de vista de seguridad se recomienda no seleccionar la opción "Guardar información de la cuenta", ya que, en caso de hacerse, se almacenarán las credenciales de acceso a la red VPN en el dispositivo móvil. Es preferible que cada vez que se establezca una conexión con una red VPN sea necesario proporcionar tanto el usuario como la contraseña.
1894. En el caso de dispositivos móviles Android multi-usuario, únicamente el usuario propietario (o usuario principal) puede modificar la configuración de las redes VPN.
1895. Una vez establecida la conexión con la red VPN, Android restringe las comunicaciones para que todo el tráfico sea cursado a través de la red VPN hasta que el dispositivo móvil se desconecte de ella, es decir, no soporta *split horizon* (conexiones simultáneas directas hacia Internet y a través de la red VPN).
1896. Una vez se ha establecido la conexión con la red VPN, aparecerá un icono con forma de llave en la barra superior de estado y se añadirá una notificación al centro de notificaciones. Para desconectarse de la red VPN, el usuario sólo debe seleccionar dicha notificación, correspondiente a la red VPN activa.
1897. Mediante una pulsación prolongada corta sobre la entrada para la red VPN disponible desde "Ajustes [Conexiones Inalámbricas y Redes] - Más... -

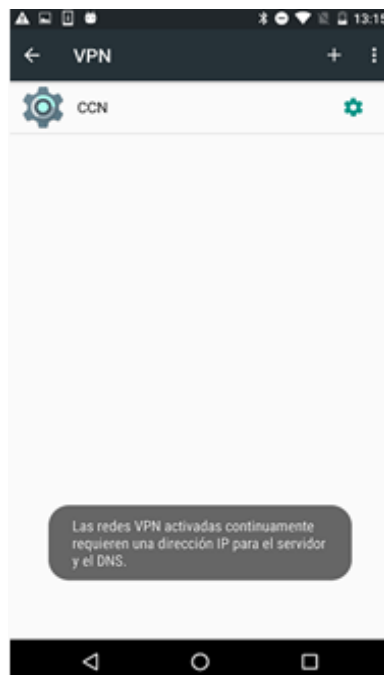
VPN", es posible editar o eliminar el perfil de la red VPN a través del botón "Olvidar":



1898. Adicionalmente, si la red VPN lo permite, Android permite marcar la opción "VPN siempre activada", forzando que no sea posible cursar ningún tráfico de red (ni recibir, ni enviar) si la conexión a la red VPN no está establecida [Ref.- 21]:



1899. Esta es la opción recomendada desde el punto de vista de seguridad, ya que permite asegurar que no se enviará tráfico no cifrado y potencialmente sensible a través de la red, ya sea la red de datos de telefonía móvil 2/3/4G o una red Wi-Fi.
1900. Para poder hacer uso de esta funcionalidad, sí es necesario seleccionar la opción "Guardar información de la cuenta" (desaconsejada salvo que se haga uso de la conexión VPN siempre activa), con el objetivo de que el dispositivo móvil disponga en todo momento de las credenciales del usuario para el acceso a la red VPN y pueda restablecer la conexión automáticamente en caso de ser necesario, como en el caso en que la conexión a la red (Wi-Fi o 2/3/4G) se haya interrumpido temporalmente.
1901. Adicionalmente, para poder hacer uso de esta funcionalidad, Android también requiere que se especifique una dirección IP (en lugar de un nombre) para el servidor o concentrador de la red VPN y para el servidor DNS:



1902. Para cambiar los ajustes de una VPN definida, se debe pulsar el icono de engranaje a la derecha del nombre.
1903. Por último, Android no dispone de soporte nativo para redes VPN basadas en OpenVPN, siendo necesario instalar una app de terceros que actúe de cliente OpenVPN para este tipo de conexiones, como por ejemplo el cliente oficial de OpenVPN para Android disponible en Google Play¹²³.

19.7. CONEXIÓN GTALKSERVICE

1904. La conexión GTalkService [Ref.- 80] [Ref.- 81], comunicación permanente a través de Internet disponible entre los dispositivos móviles Android y Google, emplea los protocolos TCP/SSL/XMPP para comunicarse periódicamente con los servidores de Google (mtalk.google.com) en el puerto TCP/5228 haciendo uso de las conexiones de datos del terminal, telefonía móvil (2/3/4G) o Wi-Fi.
1905. El servicio GTalkService es empleado para el envío de mensajes desde Google a los dispositivos móviles Android, o para la instalación o eliminación remota de apps.
1906. El protocolo empleado por GTalkService no hace uso de mecanismos adicionales de seguridad (por ejemplo, firmas criptográficas de los mensajes intercambiados) aparte del uso de HTTPS (SSL/TLS), por lo que un ataque sobre este protocolo permitiría potencialmente modificar el tráfico intercambiado y, por ejemplo, eliminar o instalar nuevas apps en el dispositivo móvil.
1907. Desde Android 5.x deja de estar disponible el código USSD `##8255##` (o `##talk##`) del teclado del teléfono en Android, que

¹²³ <https://play.google.com/store/apps/details?id=net.openvpn.openvpn>

ejecutaba el GTalk Service Monitor en Android 4.x y anteriores, por lo que ya no es posible obtener todos los detalles de ejecución de GTalkService.

1908. Por defecto, en Android 4.4.4, los detalles de este servicio estaban ocultos¹²⁴, y ni siquiera se pueden obtener los detalles de este servicio a través del botón de menú, y de la opción "Verbose logging". Para más información, consultar la versión para Android 4.x de la presente guía [Ref.-402].

1909. En versiones previas de Android, como 2.x, el *Gtalk Service Monitor* disponía de múltiples pantallas con información del servidor y puerto al que se encuentra conectado el dispositivo móvil, el identificador de usuario (Google JID) y de dispositivo móvil (Device ID), el intervalo de comprobación de la conexión (15 minutos), el estado de la conexión, un histórico de conexiones, o estadísticas de la transmisión de datos y las comunicaciones.

Nota: Los valores del "JID" (ejemplo, "usuario@gmail.com/android<CODIGO>", dónde <CODIGO> es un string de 12 valores hexadecimales) y del "Device ID" (ejemplo, "android-<CODIGO>", dónde <CODIGO> es un string de 16 valores hexadecimales) no están relacionados con el valor del "ANDROID_ID" (ver apartado "7.12. Configuración por defecto del dispositivo móvil").

20. GESTIÓN DE APPS EN ANDROID

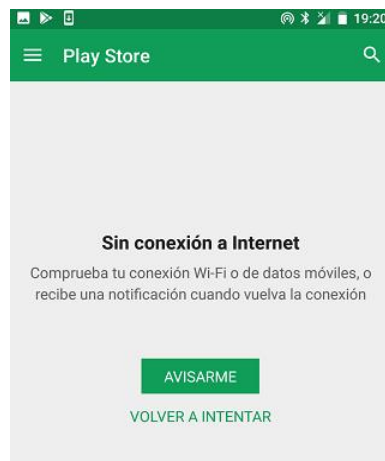
20.1. ACTUALIZACIÓN DE APPS EN ANDROID

1910. El término apps o aplicaciones móviles engloba en el presente apartado al conjunto de aplicaciones móviles de sistema o de terceros que son compatibles con Android y que pueden ser instaladas sobre este sistema operativo.

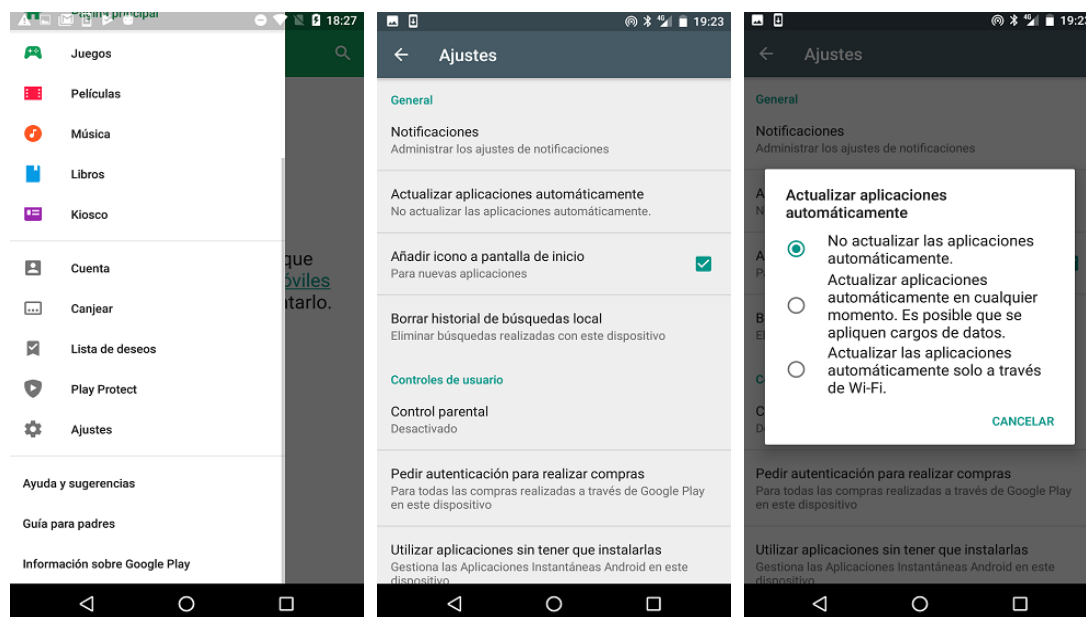
Nota: Pese a que Android permite la instalación de apps de terceros desde fuentes no oficiales (ver apartado "7.6. Instalación de apps (de orígenes desconocidos) en Android"), el análisis de las actualizaciones de apps en Android se centra en las aplicaciones móviles disponibles a través del mercado oficial de Google, denominado Google Play.

1911. La app "Play Store" no puede ser ejecutada si no se dispone de conectividad de datos (autenticada y cifrada) hacia Google Play en Internet (Wi-Fi o 2/3/4G), mostrándose el siguiente mensaje:

¹²⁴ A diferencia de como ocurría en versiones previas de Android, 2.x (ver la versión previa de la presente guía).



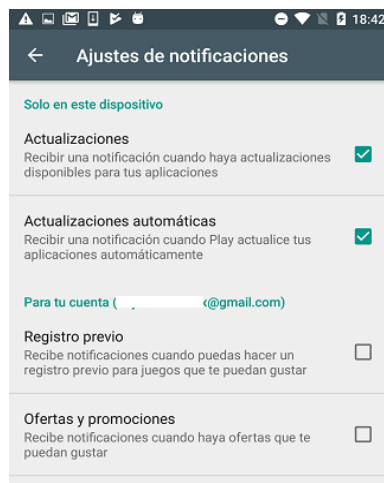
1912. No obstante, es posible que este mensaje no se muestre si la app "Play Store" guardó en cache el resultado de su última invocación exitosa. En este caso, se verá una pantalla con los nombres de las apps pero sin los iconos.
1913. Tampoco es posible ejecutar la app "Play Store" si no existe una cuenta de usuario de Google dada de alta y activa en el dispositivo.
1914. La presente guía se basa en la versión de la app (Google) "Play Store" 10.6.08, si bien alguna referencia a versiones anteriores puede estar presente para aspectos concretos.
1915. Desde la app de Google Play, denominada "Play Store", y mediante el icono de menú es posible acceder a la sección de "Ajustes". En el apartado "General" es posible configurar como se desea que se produzca el proceso de actualización de las apps instaladas desde Google Play mediante la opción "Actualizar aplicaciones automáticamente". Se dispone de la opción de que no se actualicen las apps automáticamente, que se actualicen las apps automáticamente en cualquier momento, o sólo a través de redes Wi-Fi (opción por defecto):



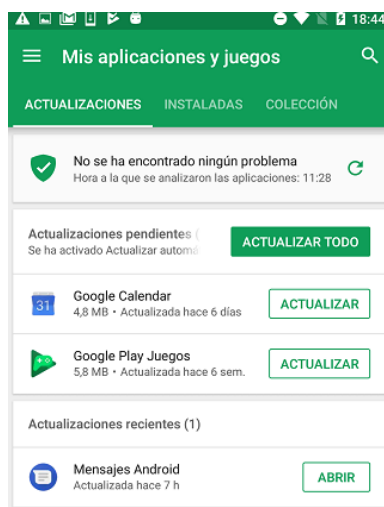
1916. En caso de estar interesados en actualizar las apps tan pronto estén disponibles, tanto a través de redes Wi-Fi como de redes de datos móviles

2/3/4G, se debe activar la opción "Actualizar aplicaciones automáticamente en cualquier momento".

1917. En Android 6.x, el acceso a Google Play (incluyendo el proceso de verificación de actualizaciones de las apps) se realiza a través de una comunicación HTTPS cifrada, pero no se emplean mecanismos de protección basados en *certificate pinning*, y, al igual que ocurría con otras aplicaciones de Google como Gmail, se acepta cualquier certificado digital de confianza que importado por el usuario al establecerse una conexión cifrada mediante TLS con los servidores de Google Play [Ref.- 79].
1918. Sin embargo, el uso que hacen las apps del almacén de certificados o credenciales en Android 7.x, descrito en el apartado "11.4. Certificate pinning y certificate blacklisting", dificulta la interceptación de tráfico HTTPS mediante el uso de un certificado propio para poder analizar, a través de técnicas MitM, las conexiones desde Google Play a los servidores de Google. El certificado del proxy de interceptación, aunque se instale en el almacén de credenciales, no será reconocido por las apps a menos que se modifique su APK para permitir la confianza en los certificados instalados por el usuario, y se instale de nuevo dicho APK manualmente. Esta técnica queda fuera del alcance de la presente guía, por lo que no se proporcionan trazas de red que permitan analizar el tráfico entre el dispositivo Android 7.x y los servidores de Google.
1919. Por defecto, la barra de notificaciones de Android mostrará un aviso sobre la disponibilidad de actualizaciones para las apps, en caso de no haber seleccionado la opción de actualización automática, ya que en ese caso las actualizaciones podrían ser llevadas a cabo sin la intervención del usuario, en concreto, si éstas no solicitan nuevos (grupos de) permisos.
1920. Mediante la selección de dicha notificación, el usuario puede identificar para qué apps se dispone de actualización y los detalles asociados.
1921. La configuración de las notificaciones está disponible desde la app "Play Store", a través del icono de menú, de la sección de "Ajustes", y mediante la opción "Notificaciones". Las notificaciones se pueden configurar a nivel de dispositivo (para informar sobre las actualizaciones que se detecten o realicen sobre el terminal) y de cuenta de Google (para que, en base a las recomendaciones de Google resultantes del análisis que se hace del perfil de la cuenta de usuario, se envíen notificaciones sobre aplicaciones y promociones que puedan interesar al usuario). Estas últimas están desactivadas por defecto, y se recomienda mantenerlas así en aras de la privacidad.

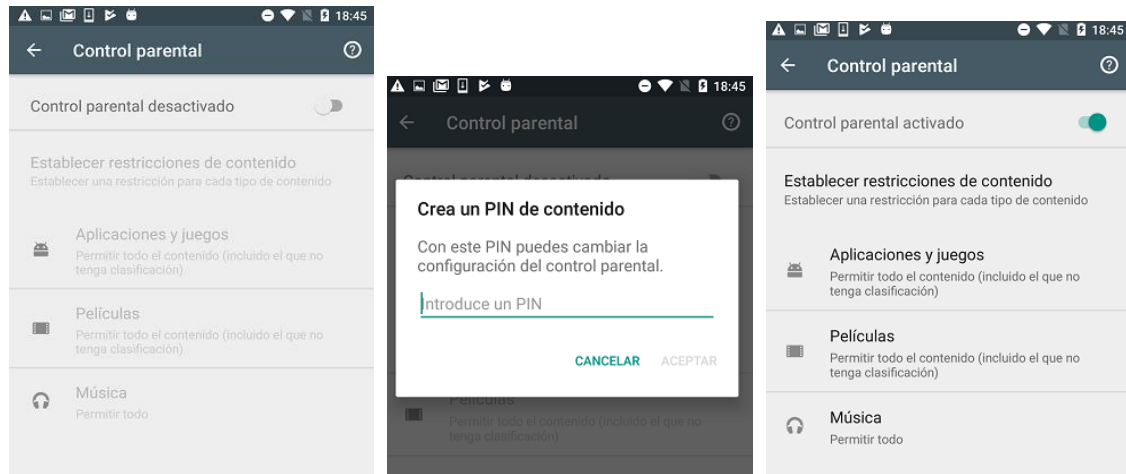


1922. La pantalla de actualizaciones está disponible desde la propia app "Play Store", mediante el icono de menú, accediendo a la sección "Mis aplicaciones y juegos" y a la pestaña "Actualizaciones". En ella se mostrarán por un lado las actualizaciones pendientes (si las hay) y, debajo, las "Actualizaciones recientes".
1923. Desde esta pantalla el usuario puede actualizar manualmente cada una de las apps, o todas ellas simultáneamente (botón "Actualizar Todo").
1924. En la pestaña "Instaladas" de "Mis aplicaciones y juegos" se puede ver también qué aplicaciones tienen actualización disponible, que además tendrán a su derecha el botón "Actualizar".

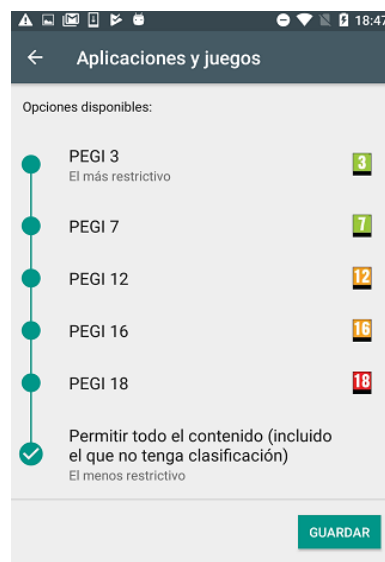


1925. Desde la misma se proporcionan detalles sobre las actualizaciones disponibles para las apps descargadas e instaladas en el dispositivo móvil, así como del resto de apps instaladas para las que no existe actualmente una actualización.
1926. Adicionalmente, Android permite establecer restricciones de contenido desde la sección "Ajustes [Controles de Usuario] Control parental" de la app "Play Store", de forma que sólo se puedan instalar las aplicaciones que cumplan con el nivel de filtro establecido. El control parental requiere de la creación de un PIN para acceder a su configuración y así evitar cambios en los controles de usuario y filtros de contenido por parte de otros usuarios

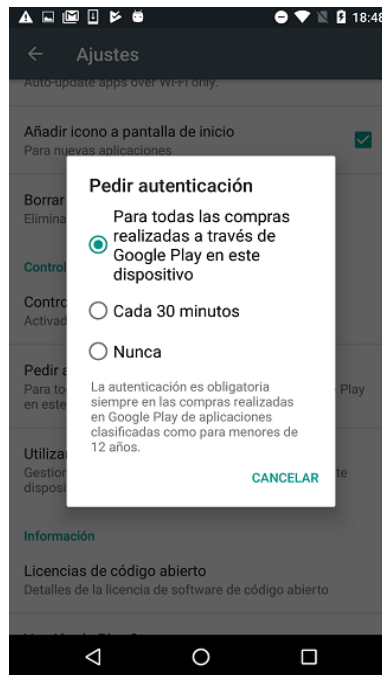
del dispositivo móvil (como por ejemplo menores de edad). Se recomienda que este PIN sea distinto del PIN de acceso al dispositivo si lo hubiere:



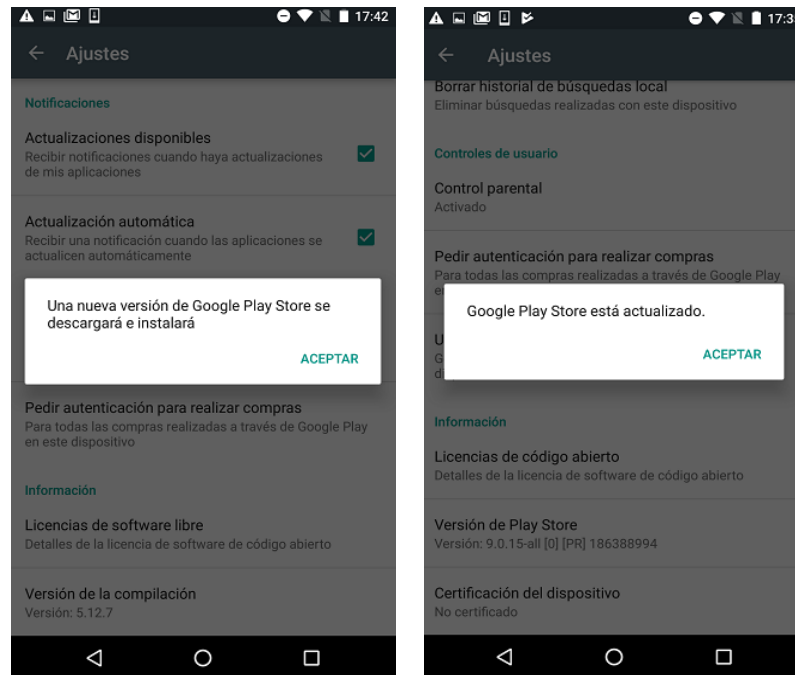
1927. Las restricciones de contenido se pueden establecer según tres categorías: películas, música y aplicaciones y juegos. Por defecto, se encuentran activadas las opciones que no restringen contenido.



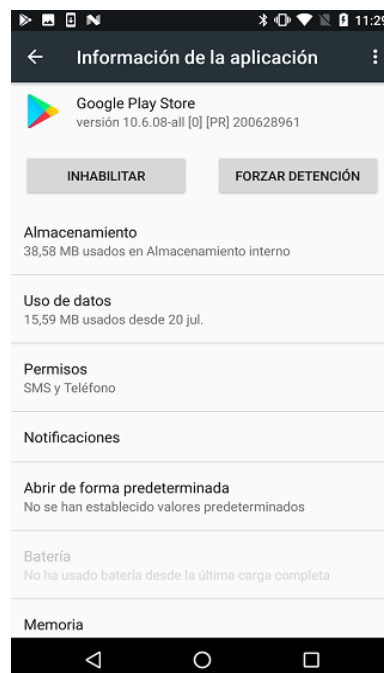
1928. Todos los detalles sobre la clasificación y configuración del control parental se pueden consultar en [Ref.- 107].
1929. Asimismo, la opción "Pedir autenticación para realizar compras" (habilitada por defecto y opción recomendada desde el punto de vista de seguridad) permite establecer controles para que, a la hora de instalar una nueva app de pago desde Google Play, se solicite autorización al usuario, junto a sus credenciales:



1930. Esta solicitud puede hacerse válida para los siguientes 30 minutos, con el objetivo de facilitar la realización de múltiples compras sin que sea necesario autenticarse para cada una de ellas, o nunca (opción desaconsejada).
1931. La app "Play Store" recibe actualizaciones silenciosas que se llevan a cabo automáticamente y sin aviso para el usuario. Se puede comprobar si hay alguna una actualización de la app "Play Store" disponible para el dispositivo entrando en los ajustes de la app "Play Store", "Ajustes - [Información] Versión de la compilación ". Al seleccionar ese campo, si hay actualizaciones disponibles, se mostrará un mensaje informativo "Una nueva versión de Google Play Store se descargará e instalará", pero ello no provoca que la descarga de la actualización se lleve a cabo inmediatamente. La única forma de forzar una actualización automática de la app "Play Store" (si se requiere una funcionalidad precisa no disponible en la versión actual) es llevar a cabo una actualización manual del fichero APK correspondiente a la app "Play Store" de un sitio web de confianza (no existe ningún sitio oficial desde el que se puedan descargar las versiones de la app "Play Store"). Cuando la versión está actualizada, se mostrará un mensaje informativo de ello.

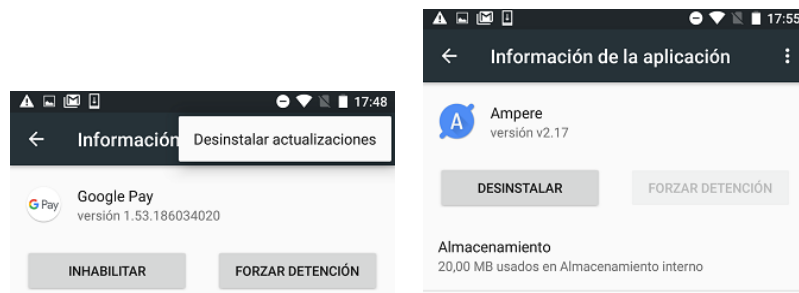


1932. La versión de la app "Play Store", al igual que de cualquier otra app instalada en Android, puede obtenerse tanto desde la propia app (menú "Ajustes [Información] - Versión de Play Store", como desde el menú "Ajustes [Dispositivo] - Aplicaciones", por ejemplo, desde la pestaña "Todas", seleccionando la app (en el ejemplo inferior, "Google Play Store") de la lista de todas las apps disponibles (por ejemplo, versión 10.6.08...):



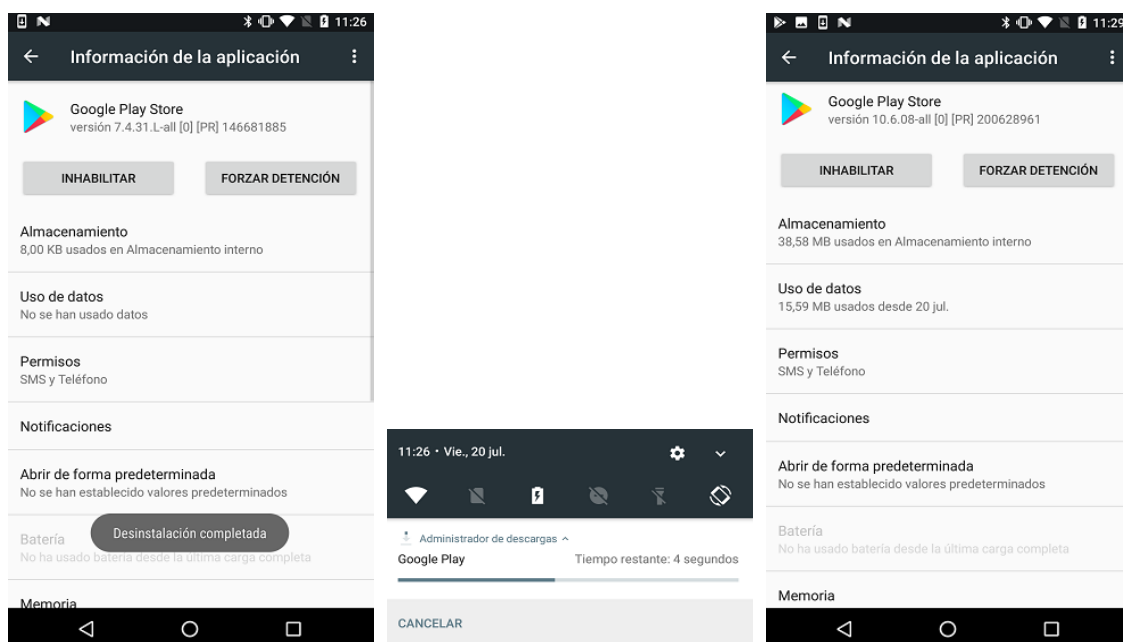
1933. La pantalla de información de una app permite tanto desinstalar la app (solo para aplicaciones que no son de sistema), mediante el botón "Desinstalar", como desinstalar todas las actualizaciones disponibles (para aplicaciones nativas del sistema y disponibles por defecto en Android), mediante el botón "[...] Desinstalar actualizaciones".

1934. Las apps de sistema no pueden ser desinstaladas completamente, sino que, al desinstalar actualizaciones, se devolverá la app a la versión que traía de fábrica.



1935. Se puede conocer qué apps son de sistema a través del menú "Ajustes - [Dispositivo] Aplicaciones - [...] Mostrar aplicaciones del sistema".

1936. Se constata en Android 7.1.2 que, aunque aparentemente es posible desinstalar las actualizaciones de la app "Play Store", tan pronto transcurren unos segundos el dispositivo recibirá la nueva actualización, sin intervención alguna del usuario, ni solicitud de confirmación por parte del sistema. Únicamente se podrá ver un símbolo de flecha de descarga en el área de notificaciones de la barra superior de estado que, al desplegarse, mostrará que la descarga de la app está en curso:



1937. Con el objetivo de disfrutar del mayor nivel de seguridad posible en el dispositivo móvil y de manera general, se recomienda disponer siempre de la última versión de todas las apps instaladas, independientemente del desarrollador. La última versión debería solucionar todas las vulnerabilidades de seguridad públicamente conocidas.

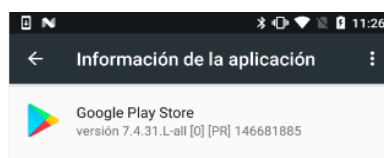
1938. Las versiones recientes de la app "Play Store" incluyen un campo "Certificación del dispositivo" cuyo propósito es determinar si el dispositivo cumple los requisitos de compatibilidad de Android [Ref.- 271]. Las pruebas realizadas durante la elaboración de la presente guía ponen de manifiesto que un dispositivo móvil puede marcarse como "No certificado" si tiene

desbloqueado el gestor de arranque (*bootloader*). A fecha de redacción de la presente guía, este estado no implica que no puedan ejecutarse instalaciones o actualizaciones del sistema operativo o de apps desde Google Play, pese a que esta posibilidad se enumera entre las consecuencias posibles que pueden aplicar en el futuro a dispositivos móviles no certificados.

1939. La información respecto a las actualizaciones de software y descargas disponibles para las apps de Android está disponible en Google Play y/o en la página web de su desarrollador.
1940. La instalación de actualizaciones de software y apps en Android, si todas las apps se han obtenido a través de Google Play, tiene asociado un proceso homogéneo, facilitando el proceso de actualización para usuarios no experimentados y aumentando el nivel de seguridad de los dispositivos móviles, al agilizarse el proceso de sustitución de software vulnerable.

20.2. ACTUALIZACIÓN AUTOMÁTICA DE APPS

1941. Android proporciona capacidades para actualizar automáticamente las apps instaladas en el dispositivo móvil tan pronto se detecte que hay disponible una nueva versión de las mismas en Google Play.
1942. Desde la app "Play Store", mediante el icono de menú es posible acceder a los "Ajustes" de Google Play. Desde la sección "General", es posible seleccionar la opción "Actualizar aplicaciones automáticamente".
1943. La versión por defecto de la app "Play Store" en Android 7.1.2 es la 7.4.31, que se actualizada automáticamente (y sin la intervención del usuario) a la última versión disponible (10.6.08 a fecha de elaboración de la presente guía).



1944. Por defecto la funcionalidad de actualización automática de las apps en Google Play está habilitada sólo para llevarse a cabo mediante conexiones de datos a través de redes Wi-Fi. Adicionalmente, es posible definir si también es posible realizar las actualizaciones automáticas de las apps a través de la conexión de telefonía móvil (2/3/4G) del dispositivo móvil, ya que esta última puede tener costes asociados.
1945. Finalmente, también es posible deshabilitar las actualizaciones automáticas de las apps por completo, siendo necesaria la intervención manual del usuario para completar la actualización de cualquier app. Se sugiere aplicar una configuración específica en base a las recomendaciones descritas posteriormente.
1946. Para ello debe modificarse la opción "Actualizar las aplicaciones automáticamente solo a través de Wi-Fi" (opción habilitada por defecto) disponible en la app "Play Store", desde el icono de menú, accediendo a los

"Ajustes" de Google Play y, en concreto, a la sección "General" (mostrada en las imágenes previas).

1947. Aunque en versiones de Android anteriores a la 5 no existía la posibilidad de desactivar las actualizaciones automáticas para apps específicas (actualizaciones selectivas), desde Android 5.x es posible seleccionar de manera individual qué aplicaciones se desea actualizar automáticamente [Ref.- 222].
1948. Desde la app "Play Store", mediante el icono de menú es posible acceder a "Mis aplicaciones y juegos", seleccionar una app concreta de las ya instaladas, pinchar en el menú [...] de la parte superior derecha y habilitar o deshabilitar la opción "Actualizar autom.". El valor de esta opción por defecto es el que aplique de la política general de actualizaciones descrita previamente.



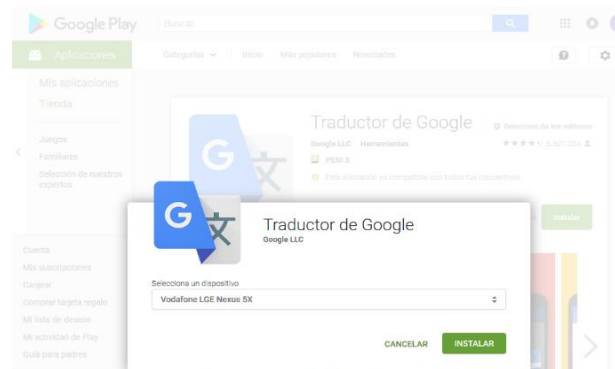
1949. Si la funcionalidad de actualizaciones automáticas ha sido habilitada, las nuevas actualizaciones serán aplicadas sin la intervención del usuario si la actualización no modifica los permisos requeridos por la app o, más recientemente, incluso si la actualización requiere disponer de nuevos permisos pero estos están englobados en un grupo de permisos previamente aprobado por el usuario para dicha app (ver apartado "7.1.4. Permisos simplificados y grupos de permisos de Google Play").
1950. Por tanto, pese a que para un usuario no avanzado sería recomendable dejar activadas las actualizaciones automáticas con el objetivo de disponer siempre de la última versión de cada app, que supuestamente solucionará las vulnerabilidades conocidas públicamente, debido al nuevo modelo de permisos simplificados de Android, es preferible llevar a cabo las actualizaciones de las apps manualmente.
1951. Se recomienda por tanto no dejar habilitada la opción de actualizaciones automáticas, e independientemente del perfil del usuario del dispositivo

móvil, forzar al usuario a revisar manualmente los nuevos permisos solicitados por cada una de las actualizaciones recibidas para todas y cada una de las diferentes apps instaladas en el terminal.

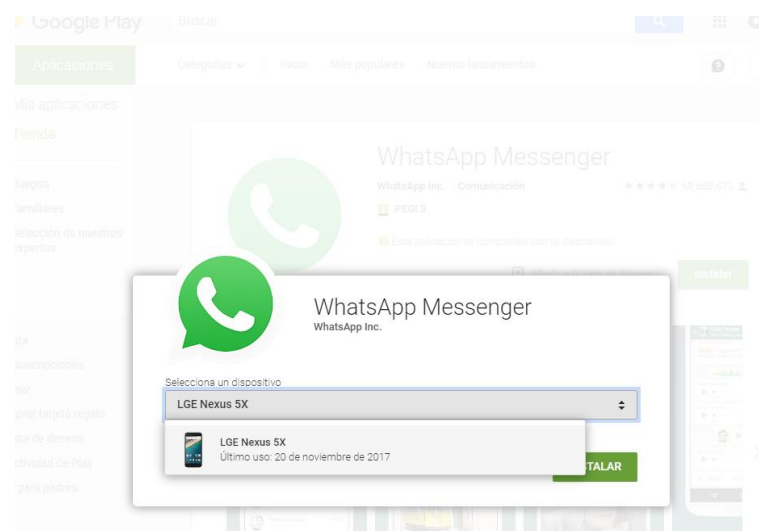
1952. Especialmente, los usuarios avanzados deben estar interesados en no habilitar esta opción para poder revisar y analizar minuciosamente las actualizaciones disponibles, y sus "nuevos" permisos, antes de que éstas sean instaladas.
1953. En caso de que los permisos de la app hayan cambiado con la actualización, el proceso de actualización manual mostrará los nuevos permisos requeridos por la nueva versión de la app, que deberán ser aprobados por el usuario durante la instalación de la actualización de la app. Sin embargo, recientemente, si los nuevos permisos requeridos por la app están englobados en un grupo de permisos previamente aprobado por el usuario para dicha app, la actualización podrá ser instalada automáticamente sin la intervención del usuario.

20.3. INSTALACIÓN REMOTA DE APPS

1954. En mayo de 2010 Google presentó la versión web de Google Play [Ref.- 4], que permite la consulta e instalación de apps en los dispositivos móviles Android desde otros ordenadores mediante un navegador web, en lugar de emplear la app de acceso al mercado oficial ("Play Store") desde los propios dispositivos móviles.
1955. Una vez se adquiere una app a través de la versión web de Google Play desde el mercado oficial de apps, Google se encarga de instalarla en el dispositivo móvil a través de una conexión persistente disponible en los dispositivos móviles basados en Android a través de Internet (ver apartado "19.7. Conexión GTalkService").
1956. La conexión permanente a través de Internet disponible entre los dispositivos móviles Android y Google se basa en el GTalkService [Ref.- 80]. Para la instalación remota de apps se emplea un mecanismo de envío de mensajes propio de Google y Android, y en concreto en este caso, el mensaje denominado INSTALL_ASSET [Ref.- 81].
1957. Antes de llevar a cabo la instalación de cualquier nueva app, se recomienda revisar y analizar exhaustivamente los permisos solicitados por la misma sobre el dispositivo móvil.
1958. El acceso a Google Play se realiza mediante la cuenta de usuario de Google, por lo que el acceso no autorizado por parte de un atacante a dicha cuenta permitiría la instalación de apps en el dispositivo móvil asociado del usuario [Ref.- 8].
1959. Tras seleccionar la app en la web de Google Play, el usuario puede seleccionar en cuál de los diferentes dispositivos móviles vinculados a la cuenta de usuario de Google se desea realizar su instalación:



1960. Tras confirmar su instalación en el dispositivo móvil seleccionado, se obtiene la confirmación a través de la página web de Google Play, y en el dispositivo móvil el proceso de instalación se lleva a cabo automáticamente, conllevando únicamente un aviso en la barra de notificaciones del dispositivo (y la aparición del icono de la app en alguna de las pantallas de inicio y/o en el *Launcher*):



1961. Es importante reseñar que, aunque la instalación no pueda llevarse a cabo en el preciso momento en que se solicita, tan pronto se pulse el botón "Aceptar" el interfaz de Google Play mostrará esa app como "Instalada". Esto ocurre incluso aunque en el momento de solicitarse la instalación el dispositivo móvil no tenga conexión a los servidores de Google. La petición queda encolada en la cuenta de Google del usuario y, tan pronto se restablezca la conexión con los servidores, se iniciará la descarga e instalación de la app.

1962. En el pasado, la instalación remota de apps desde la web de Google Play de forma no autorizada podía suceder si un potencial atacante obtenía los identificadores de sesión empleados en el acceso web a Google Play del usuario [Ref.- 78]. Google resolvió el problema modificando la cookie "androidmarket" para que incluyese el flag *HttpOnly*, con lo cual el identificador de sesión solo es enviado por el navegador cuando hace una petición, por lo que no está disponible para ataques de XSS.

20.4. ELIMINACIÓN REMOTA DE APPS

1963. Los dispositivos móviles basados en Android disponen también de capacidades por parte del vendedor, Google/Android, para eliminar apps de los mismos de forma remota.
1964. Este mecanismo fue diseñado como medida de seguridad en situaciones de emergencia para poder proteger a los usuarios en caso de que se produzca una distribución masiva de apps dañinas o maliciosas concretas.
1965. En junio de 2010 Google empleó estas capacidades por primera vez [Ref.- 5] [Ref.- 81] para eliminar una app creada como prueba de concepto que pretendía demostrar la facilidad para introducir apps maliciosas en los mercados de aplicaciones de varios fabricantes de dispositivos móviles, incluyendo Android Market (nombre empleado para Google Play en aquel momento).
1966. Posteriormente, Google ha hecho uso de esta funcionalidad en (al menos) otra ocasión para eliminar otras apps maliciosas introducidas en Android Market. Por ejemplo, en marzo de 2011, se vieron afectadas unas 50 aplicaciones infectadas con malware bajo el nombre de DroidDream que fueron instaladas en más de 200.000 dispositivos móviles (cifras no oficiales). Estas apps hacían uso de vulnerabilidades conocidas en versiones de Android previas a la versión 2.2.2, y específicamente, técnicas empleadas habitualmente para conseguir acceso como root a dispositivos móviles Android (proceso conocido como *root*, *rooted* o *rooting* del dispositivo Android).
1967. Esas capacidades permiten a Google no sólo eliminar las apps maliciosas de Google Play y suspender las cuentas de los desarrolladores sospechosos, sino también eliminar las apps de los dispositivos móviles de los usuarios donde han sido instaladas, notificando al usuario sobre las acciones realizadas. Tal y como se indica en [Ref.- 6], *"Nota: Si Google determina que la actualización de una aplicación permite solucionar una vulnerabilidad de seguridad crítica, es posible que actualicemos ciertas aplicaciones independientemente de la configuración de actualización establecida en la aplicación o en tu dispositivo. Para obtener más información, consulta las Condiciones de Servicio de Google Play."*
1968. La conexión persistente entre Google y los dispositivos móviles basados en Android también podría permitir la instalación remota de apps, y se emplea en la instalación de apps adquiridas desde el interfaz web de Google Play (detallada en el apartado previo).

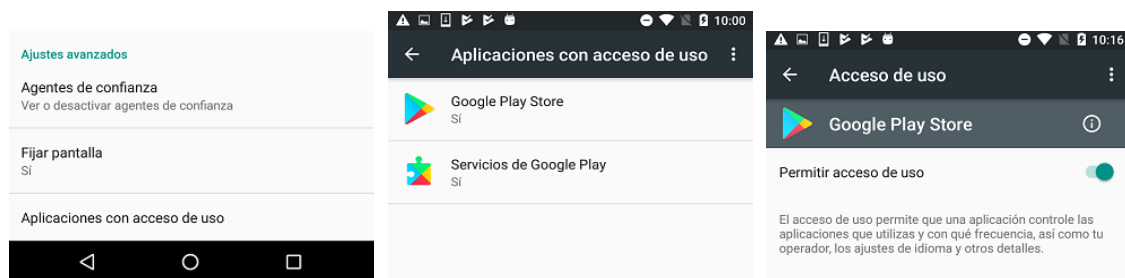
20.5. ESTADÍSTICAS DETALLADAS DE USO DE LAS APPS

1969. Tradicionalmente, una de las técnicas de ataque a aplicaciones móviles (especialmente de banca on-line) es la que se conoce como "overlay attack" (ataque de superposición). Este ataque consiste en suplantar la ventana activa de una app superponiendo una ventana maliciosa para recabar datos sensibles del usuario (credenciales, tarjetas de crédito, etc.). Para

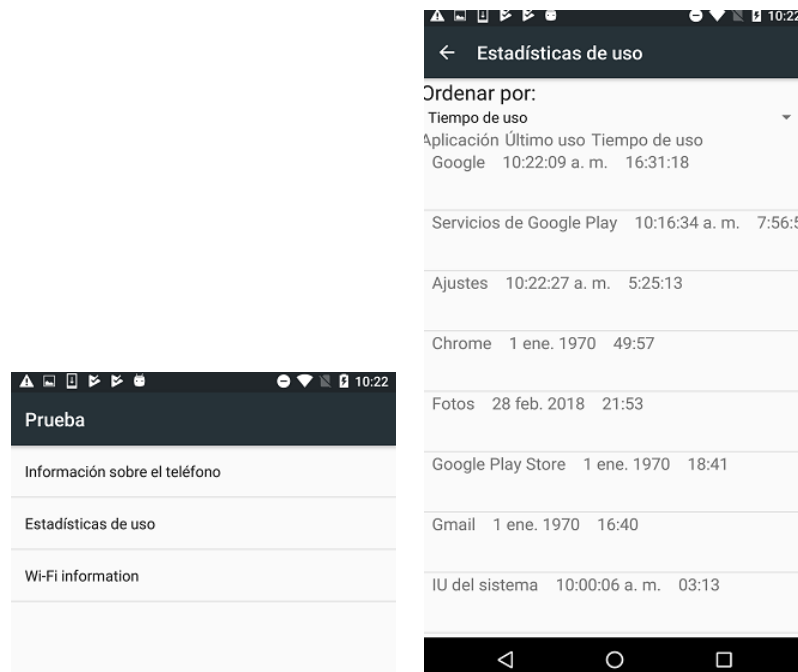
implementar este tipo de ataque, un elemento clave es poder determinar qué aplicación está ejecutando en primer plano.

1970. Android 5.0 Lollipop afrontó esta amenaza introduciendo la API *UsageStatsManager* en sustitución de la API *getRunningTasks()*. La API *UsageStatsManager* introdujo un permiso de sistema, denominado `PACKAGE_USAGE_STATS`, sin el cual las apps no dispondrán de capacidad para recabar estadísticas detalladas de uso de todas las apps disponibles en el dispositivo móvil.

1971. Para otorgar o revocar este permiso, el usuario puede acceder a "Ajustes - [Personal] Seguridad - [Avanzado] Aplicaciones con acceso de uso". También es posible modificar este permiso mediante "Ajustes - [Dispositivo] Aplicaciones - [icono de engranaje] [Opciones avanzadas] - Acceso especial - Acceso de uso":



1972. Para recabar estadísticas de uso de las apps, Android dispone de la función experimental "Estadísticas de uso", accesible mediante la marcación en la app "Teléfono" del código `*##4636#*##`:

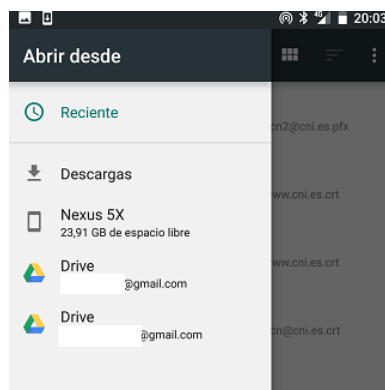


1973. La ventana que se despliega contiene el nombre de la app, la fecha de último uso y el tiempo de uso.

1974. Aunque existen diversas apps que permiten obtener estadísticas más detalladas sobre el uso de otras apps, no se recomienda hacer uso de ellas.

20.6. ACCESO Y ALMACENAMIENTO DE DATOS DESDE APPS

1975. Android 4.4 introdujo el *Storage Access Framework* (SAF), un entorno que simplifica el acceso, edición y almacenamiento de documentos, imágenes, vídeos y otros ficheros por parte de las apps de Android, con la capacidad de emplear múltiples proveedores de almacenamiento o proveedores de documentos (*Document Providers*) [Ref.- 165].
1976. Con Android 5.x (Lollipop), una nueva API (nivel 21) permite acceso total a los directorios y ficheros de las tarjetas SD, otorgando acceso completo, almacenamiento automático y mejorando la seguridad. Las apps pueden seleccionar un subárbol de directorio entero y proporcionar acceso de lectura y escritura a todos los documentos contenidos en él, sin requerir la confirmación del usuario para cada uno de ellos.
1977. Android 5.x añadió el *intent* (elemento o mensaje de comunicación entre componentes de una aplicación o de aplicaciones distintas) ACTION_OPEN_DOCUMENT_TREE, que puede ser declarado por una app para recibir un directorio de cualquier componente "DocumentProvider", incluyendo el almacenamiento compartido del dispositivo. La app podrá entonces crear, modificar y borrar ficheros y subdirectorios bajo el árbol recibido, incluso de forma persistente entre reinicios del dispositivo móvil.
1978. Los servicios o proveedores de almacenamiento pueden ser tanto locales (memoria interna del dispositivo) como pertenecer a servicios de almacenamiento "en la nube" (como por ejemplo Google Drive, One Drive, Dropbox, Box, etc.), y las apps pueden obtener, modificar, borrar, crear y guardar ficheros o documentos en ellos de manera homogénea a través de un interfaz estándar y unificado, así como acceder y compartir documentos proporcionados por otras apps existentes en el dispositivo móvil a través de SAF:



1979. Estas nuevas capacidades influyen principalmente en las capacidades disponibles para los desarrolladores de apps para Android, permitiéndoles gestionar múltiples documentos o hacerlo individualmente, establecer permisos sobre estos, y gestionar la autenticación del usuario en los servicios de almacenamiento en la nube antes de realizar operaciones con los datos a guardar o gestionar.
1980. Para el usuario del dispositivo móvil, los detalles de dónde y cómo se almacenan los documentos por parte de las apps son prácticamente

transparentes, por lo que únicamente debe prestar especial atención desde el punto de vista de seguridad a si desea almacenar los documentos solo localmente, o en algún servicio "en la nube".

1981. En dispositivos multi-usuario, el almacenamiento interno no se comparte entre apps, sino que los datos disponibles para una app dependen del usuario que la ejecuta.

1982. Las apps pueden almacenar los datos:

- En el almacenamiento interno (o *sandbox*) asociado a la app, al cual sólo esa aplicación tiene acceso salvo que se haya realizado el proceso de *rooting* al dispositivo móvil (o que se explote una vulnerabilidad). Toda aplicación que se instala en Android 7.x recibe acceso a su directorio `"/data/data/<appname>"` (donde *appname* es el nombre del paquete asociado a la app). Para controlar el acceso al directorio de la app, se emplea el mecanismo estándar de permisos de ficheros y directorios de Linux, ya que cada app que se instala recibe un identificador de usuario (UID) y de grupo (GID) únicos.
- En el almacenamiento compartido (o externo): para ello, con el nuevo modelo de permisos introducido por Android 6.x (ver apartado "7.1. Modelo de permisos de Android"), la app deberá solicitar al usuario permiso sobre el almacenamiento compartido (o tarjeta SD externa). Si se le concede, la app dispondrá de permisos globales de lectura y escritura en el mismo.

20.7. ALMACENAMIENTO DE CREDENCIALES E INFORMACIÓN SENSIBLE EN APPS

1983. El almacenamiento de credenciales e información sensible en Android es uno de los elementos críticos que debería ser tenido en cuenta por los desarrolladores de apps.

1984. Adicionalmente, por defecto el almacenamiento de datos se lleva a cabo sin cifrar, es decir, las credenciales de acceso a diferentes servicios y otra información sensible será almacenadas en ficheros o bases de datos (SQLite) en claro [Ref.- 109], por ejemplo, dentro del *sandbox* de la app bajo el directorio `"/data/data/<appname>"`, o alguno de sus subdirectorios. El desarrollador de la app puede decidir donde almacenar los datos: preferencias compartidas, ficheros, bases de datos SQLite, etc.

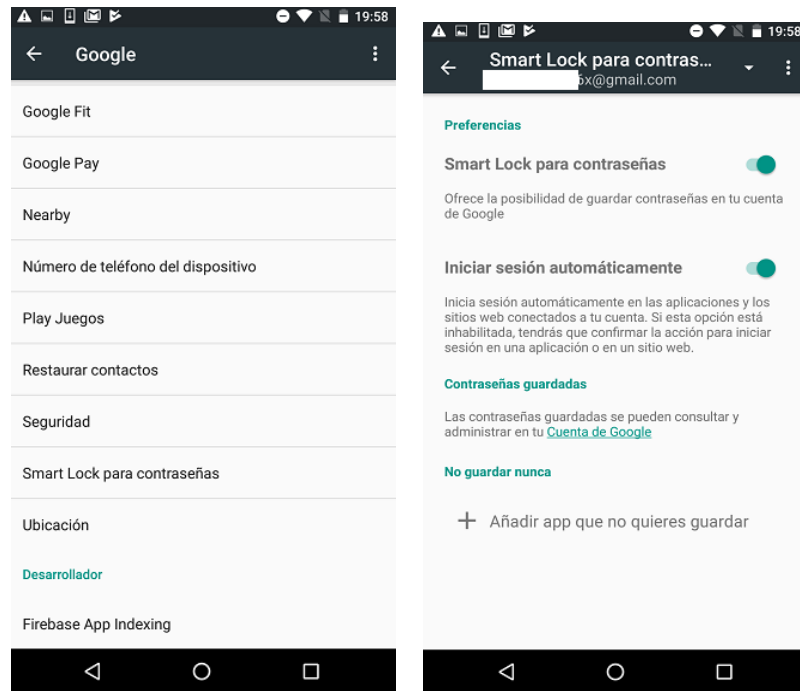
1985. Si un atacante obtiene acceso físico al dispositivo móvil, podría seguir el proceso de *rooting* (o explotar una vulnerabilidad) y disponer potencialmente de acceso a las credenciales almacenadas sin cifrar.

1986. Se recomienda por tanto que los desarrolladores de apps almacenen los datos cifrados empleando una clave que no sea accesible directamente por la app, por ejemplo, almacenándola en un KeyStore (ver apartado "11.1. Android Keystore System: Almacenamiento de credenciales") protegido por una contraseña que debe ser proporcionada por el usuario y que nunca es almacenada en el dispositivo móvil [Ref.- 184].

1987. Adicionalmente, se recomienda hacer uso de los mecanismos de cifrado del dispositivo móvil disponibles por defecto desde Android 3.0 (ver apartado "10.1. Cifrado del dispositivo móvil").
1988. Algunas apps soportan el almacenamiento de sus credenciales en gestores de contraseñas, uno de los cuales es el proporcionado por Google denominado "Smart Lock para contraseñas", que se describe en el apartado "20.8. Servicio de contraseñas de Google Smart Lock".
1989. La conveniencia o no de usar un gestor para contraseñas debe ser evaluada cuidadosamente por el usuario. Aunque la recomendación general desde el punto de vista de seguridad es mantener las contraseñas en un lugar externo al dispositivo móvil, puede resultar más conveniente para el usuario recurrir a un gestor de contraseñas que optar por almacenar las credenciales en el almacenamiento interno del dispositivo sin cifrar.

20.8. SERVICIO DE CONTRASEÑAS DE GOOGLE SMART LOCK

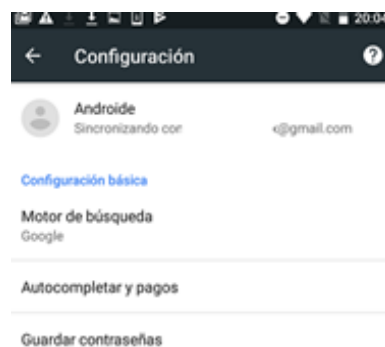
1990. Además del servicio Smart Lock para mantener dispositivos Android desbloqueados (ver apartado "9.5. Desbloqueo mediante Smart Lock"), Google incorporó con la actualización a la versión 7.5 de Google Play Services (GPS) un gestor de contraseñas referenciado como "Smart Lock para contraseñas", destinado a almacenar y sincronizar las contraseñas de sitios web accedidos con Chrome y otras aplicaciones desde los dispositivos móviles.
1991. Smart Lock para contraseñas forma parte del ecosistema "Google Identity Platform" [Ref.- 244], cuyo objetivo es la gestión segura de la identidad de los usuarios.
1992. La sincronización de datos se realiza a través de la cuenta de Google del usuario, por lo que las contraseñas almacenadas a través del gestor Smart Lock estarán disponibles en todos los dispositivos en los que se haya iniciado sesión con dicha cuenta de usuario [Ref.- 243].
1993. Cuando se visita por primera vez un sitio web o se utiliza una app que emplea esta funcionalidad y que requiere introducir credenciales, si Smart Lock para contraseñas está activo, se mostrará una ventana solicitando permiso para almacenar las credenciales. Si se acepta, la próxima vez que se visite el sitio web o se ejecute la app desde cualquier otro dispositivo que haya iniciado sesión en la cuenta de Google del usuario, no se solicitará identificación, sino que Smart Lock proporcionará automáticamente las credenciales para ingresar en dicha web o app.
1994. A nivel de app, los desarrolladores pueden integrar Smart Lock en su aplicación móvil utilizando la API "Credentials" [Ref.- 245].
1995. La activación del servicio "Smart lock para contraseñas" se realiza a través de "Ajustes - [Personal] Google - [Servicios] Smart lock para contraseñas" [Ref.- 238]. Las opciones disponibles son las siguientes:



- [Preferencias] Smart Lock para contraseñas: esta opción habilita/deshabilita el servicio.
 - [Preferencias] Iniciar sesión automáticamente: si se deshabilita, se solicitará al usuario confirmación para iniciar sesión; si se habilita, el inicio de sesión tendrá lugar sin intervención del usuario.
 - [Contraseñas guardadas]: redirige a la cuenta del usuario actualmente activa en el dispositivo móvil para poder consultar la información de credenciales guardada.
 - [No guardar nunca]: permite elegir qué apps de las instaladas por el usuario no podrán hacer uso del servicio y, por tanto, no verán sus credenciales almacenadas.
1996. Si Smart Lock para contraseñas está activado (es el modo por defecto), Chrome hará uso de él para guardar las credenciales de los sitios web para los que el usuario acepte esta acción.
1997. Si Smart Lock para contraseñas está desactivado, pero se ha iniciado sesión en la cuenta de Google del usuario, Chrome también almacenará las credenciales de los sitios web en "https://passwords.google.com".
1998. Las contraseñas de páginas web y aplicaciones Android guardadas en este gestor, así como las opciones de configuración actual asociadas a él, se pueden consultar desde "https://passwords.google.com".
1999. A través de este sitio web, se pueden eliminar entradas, pero también visualizar las contraseñas e identificadores de usuario de los sitios webs para los que se aceptó el almacenamiento:



2000. La sección "Contraseñas guardadas" sólo contendrá información sobre las credenciales de sitios web si la configuración del navegador Chrome tiene habilitada la opción de "Guardar contraseñas" dentro de su configuración:



2001. Desde el punto de vista de seguridad, se debe evaluar la confianza en el servicio antes de optar por utilizar Smart Lock para contraseñas (o cualquier otro gestor de contraseñas), debido a su criticidad.
2002. En general, se desaconseja hacer uso de las opciones que implican llevar a cabo un inicio de sesión automático, que puede provocar que un tercero no autorizado que consiga acceso temporal al dispositivo móvil sin bloquear pueda acceder a apps y servicios suplantando al propietario del dispositivo.

20.9. ENLACES Y REFERENCIAS WEB DESDE LAS APPS

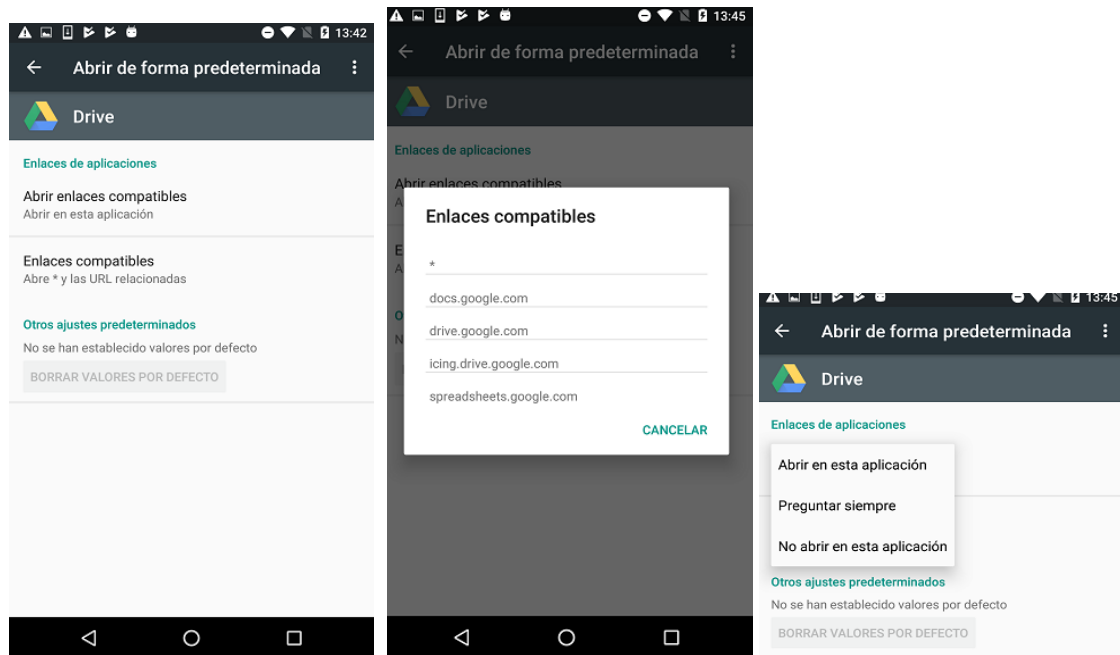
2003. En relación directa con las apps, Android 6.x añade capacidades de gestión sobre los enlaces (URLs o *links*) asociados o registrados para cada app, y que se denominan "(Android) App Links".
2004. Es decir, los "App Links" son un tipo de enlace que las apps pueden incluir en su configuración para que, ante una solicitud de apertura de un enlace de ese tipo, se ejecute una actividad específica de la app en lugar de que se abra el enlace dentro de un navegador web. Esta funcionalidad permite que ciertos enlaces (o URLs) sean gestionados directamente por una

app instalada en el dispositivo móvil, en lugar de por el servidor web asociado, en su versión para móviles, a través del navegador web [Ref.- 604]. La funcionalidad aplica a enlaces web asociados a los protocolos HTTP y HTTPS.

2005. La gestión automática de enlaces requiere que tanto el desarrollador de la app como el administrador del sitio, servidor o aplicación web trabajen conjuntamente. La app debe indicar que ella es la encargada de gestionar ese tipo de enlaces a través de su fichero *manifest* (mediante los filtros de *intents*), y la aplicación web debe indicar qué app gestionará sus enlaces mediante un fichero de enlaces a contenidos digitales (Digital Asset Links [Ref.- 269], ".well-known/assetlinks.json") en formato JSON, verificando así la asociación entre ambas.
2006. Los "Digital Asset Links" establecen pues qué enlaces o URLs serán abiertos por qué app, en función del servidor web o dominio destino, y definiendo el proceso de autorización requerido por parte del usuario cuando se abre un enlace registrado.
2007. El objetivo de la nueva funcionalidad es que se pueda establecer una app por defecto encargada de gestionar un tipo de enlace HTTP o HTTPS determinado, evitando así que cada vez que el usuario pulse en un enlace de ese tipo, aparezca una ventana de diálogo preguntando cómo desea abrir dicho enlace: navegador web, app, etc.
2008. Desde "Ajustes - [Dispositivo] Aplicaciones - [<Nombre de la App>] Abrir de forma predeterminada" el usuario dispone de la capacidad de gestionar qué apps serán las asociadas por defecto a manejar qué enlaces. La app encargada por defecto de un tipo de enlaces también se puede seleccionar por parte del usuario mediante la opción "Siempre", al ser consultado a través de la ventana de diálogo correspondiente, la primera vez que se abrió un enlace de ese tipo.

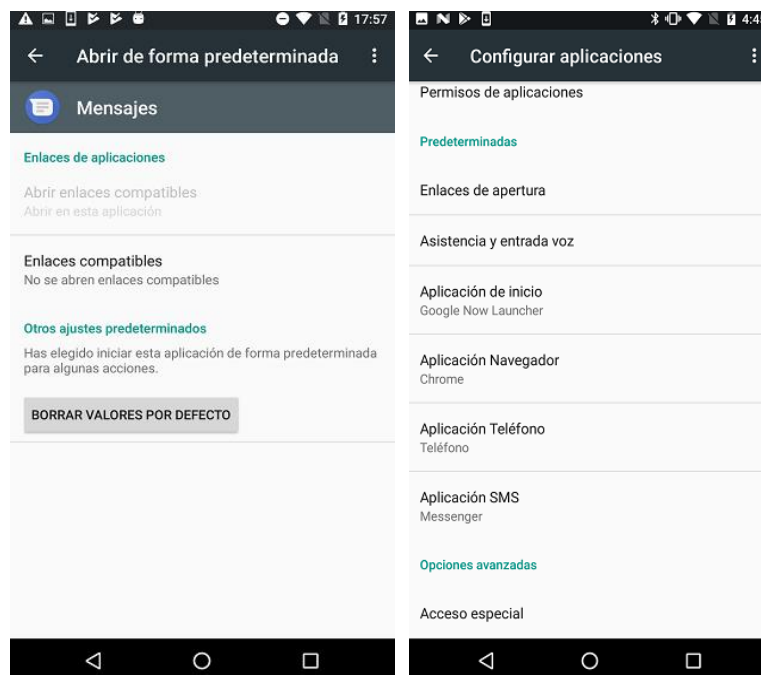
Nota: Los desarrolladores interesados en la creación de apps para Android disponen de todos los detalles e información en el portal para desarrolladores *Google Play Developer Console* (<https://play.google.com/apps/publish>) [Ref.- 7].

2009. La opción "Abrir enlaces compatibles" permite definir si las URLs que la app declaró como "Enlaces compatibles" se abrirán automáticamente, no se abrirán o se consultará al usuario:



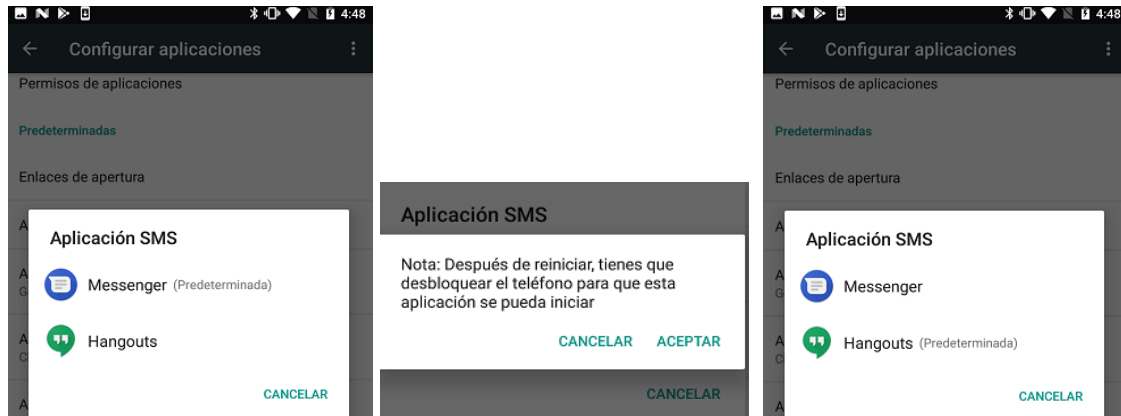
2010. La opción "Enlaces compatibles" son los sitios web que la app declaró en su fichero *manifest* (y en el fichero *assetlinks.json* asociado al dominio) como propios para lanzar una actividad concreta dentro de ella.

2011. La opción "Otros ajustes predeterminados" informa de si la app se ha seleccionado como aplicación para ejecutarse ante un determinado evento. Sin embargo, este campo sólo indica si la app tiene algún ajuste de ese tipo, pero no concretamente cuál. Para consultarlo, se debe acceder a "Ajustes - [Dispositivo] Aplicaciones - [rueda de configuración] [Predeterminadas]" y examinar la lista de apps.



2012. En caso de haberse seleccionado como predeterminada una app para una acción, si se desea cambiarla, se puede elegir entre los siguientes métodos:

- Utilizar el botón "Borrar valores por defecto" y lanzar la acción correspondiente para que el sistema pregunte de nuevo al usuario que app desea que se emplee de forma predeterminada.
- Pinchar sobre la app predeterminada para una categoría y elegir la app que se desee emplear en su lugar:



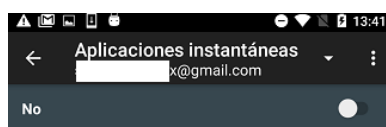
2013. El modelo empleado por Android para determinar quién se encargará de manejar un enlace se basa en los siguientes tres criterios o escenarios: si el usuario ha designado una app por defecto para dicho enlace, Android se lo enviará, y ésta se encargará de abrirlo. Si no hay establecida ninguna preferencia o asociación para ese tipo de enlace, y solo hay una app instalada para gestionar ese enlace, dicha app recibirá la petición. Si no hay establecida ninguna preferencia o asociación para ese tipo de enlace, y hay varias apps instaladas que pueden gestionar ese enlace, Android mostrará una ventana de diálogo preguntando al usuario qué app debe recibir la petición.

20.10. INSTANT APPS

2014. Las "Android Instant Apps" son aplicaciones nativas Android que se ejecutan de forma instantánea sin requerir instalación en el dispositivo móvil. En lugar de ello, el portal web que distribuye la app proporcionará una URL que, bajo demanda, cargará sólo la parte de una app necesaria para ejecutar una determinada acción.
2015. Uno de los principales beneficios de las Instant Apps es no tener que instalar aplicaciones sólo para realizar una tarea específica desde una web (por ejemplo, realizar un pago, pedir un taxi, etc.), sino aprovechar una app propia disponible a través de un enlace web al portal que proporciona el servicio, desde el que se cargará la app requerida.
2016. Al evitar la instalación/desinstalación de apps que sólo se requieren puntualmente, se ahorra espacio en el dispositivo móvil.
2017. Las Instant Apps permiten a los desarrolladores segmentar su app en módulos que pueden ejecutarse independientemente dentro de un sandbox, de forma que sólo se descargará el código necesario para una determinada acción, con el consiguiente ahorro de RAM en el dispositivo móvil. Por ejemplo, un usuario está de viaje en el extranjero y necesita pedir comida

rápida. En lugar de instalar la aplicación de una cadena de restauración, el portal web puede ofrecer un enlace que lance la funcionalidad de búsqueda de menú (en lugar de toda la app); además, se puede integrar con Android Pay para que el usuario pueda completar el pago sin requerir más acción por su parte.

2018. Según Google, el esfuerzo a nivel de desarrollo para transformar una app tradicional en una Instant App es mínimo, pues no requiere rediseñar la app.
2019. Aunque las Instant Apps se presentaron en el Google I/O de 2016 [Ref.- 240], estas son compatibles con dispositivos desde Android 4.1 (nivel de API 16) y Google Play Services [Ref.- 239].
2020. Son varios los propósitos de las Instant Apps: por un lado, pretenden mejorar la experiencia del usuario aprovechando su integración con las funciones de Google Play Services (como la ubicación, la identidad, los pagos, etc.); por otro, permitir a los desarrolladores de apps que las mismas sean usadas sin necesidad de instalarse físicamente en el dispositivo móvil.
2021. Desde el punto de vista de seguridad, surgen varias cuestiones a tener en cuenta: cuál será la gestión de permisos de las Instant Apps (especialmente en dispositivos anteriores a Android 6.x), cómo saber si se puede confiar en una Instant App que se abre desde un determinado enlace web, cómo saber si una Instant App cumple con políticas de seguridad corporativas, qué potenciales riesgos introduce la propia traducción de un enlace web a código ejecutable... [Ref.- 241] [Ref.- 242]. Por el momento, Google no ha elaborado ningún documento técnico abordando estas cuestiones, por lo que se recomienda tener prudencia a la hora de hacer uso de esta funcionalidad.
2022. Si no se desea hacer uso de este tipo de aplicaciones, se puede deshabilitar su uso a través de "Ajustes - [Personal] Google - [Servicios] Aplicaciones instantáneas" y mover el interruptor a la posición "No".



Activa las Aplicaciones Instantáneas para iniciar aplicaciones al instante sin tener que instalarlas. [Más información](#)

2023. La versión 1.1 del SDK de Instant Apps añade un contexto persistente después de la instalación de una Instant App, permitiendo a la app completa acceder a los datos de su versión "Instant App". De este modo, si el usuario instala una versión completa tras haber ejecutado una Instant App, podrá recuperar elementos de configuración y archivos creados por ella.

2024. Adicionalmente, de cara a los desarrolladores, esta versión 1.1 del SDK incluye APKs de configuraciones, de forma que se disponga de distintas variantes según la arquitectura (por ejemplo, ARM o ARM64) o la resolución de la pantalla del dispositivo móvil, de forma que la app descargada tenga un tamaño menor.

21. COPIAS DE SEGURIDAD

2025. Mantener una buena política de backup (copia de seguridad) es un elemento indispensable, tanto de forma periódica como antes de proceder a una actualización de sistema operativo.
2026. Se ofrecen dos alternativas principales: utilizar las capacidades de Android o recurrir a los servicios de Google Drive. Desde el punto de vista de la privacidad, se recomienda utilizar una política de backup local.

21.1. COPIA DE SEGURIDAD LOCAL

2027. Desde la versión 4.0 de Android (*Ice Cream Sandwich*) se dispone de capacidades locales para realizar copias de seguridad de los contenidos del dispositivo móvil en el ordenador del usuario [Ref.- 93], sin necesidad de haber realizado el proceso de *rooting* sobre el mismo.
2028. Las copias de seguridad se llevan a cabo a través de ADB, por lo que es necesario disponer de las capacidades de desarrollo y depuración a través de USB habilitadas (ver apartado "13.3. Opciones de desarrollo y depuración USB").
2029. Desde el punto de vista de seguridad se recomienda habilitar estas capacidades únicamente en el momento de llevar a cabo la copia de seguridad, y volverlas a deshabilitar inmediatamente después.
2030. En concreto, el comando "*adb backup*" permite realizar la copia de seguridad, pudiéndose especificar el fichero donde se almacenará la misma (mediante la opción "-f", y por defecto en "backup.ab"). Este comando dispone de diferentes opciones para definir qué contenidos serán añadidos a la copia de seguridad, incluyendo los datos de las apps, tanto de sistema (o apps existentes por defecto) como de terceros, los propios ficheros APK de las apps, los contenidos del almacenamiento compartido (o tarjeta SD), etc. Es también posible especificar que se realice la copia de seguridad sólo de ciertas apps.
2031. Se han observado casos en los que, al utilizar una versión reciente del binario "adb" incluido en las "Platform Tools" del SDK de Android para guardar un backup de una versión de Android 4.x, la invocación del comando "adb backup" da como resultado un fichero de extensión ".ab" vacío o con escaso tamaño. Parece que el problema se debe a que el dispositivo hace uso de una versión antigua del protocolo ADB, en la que los argumentos del comando "adb backup" eran escapados de manera diferente, por lo que es necesario hacer uso de una versión previa del comando "adb" en el ordenador utilizado para realizar la copia de seguridad. Al hacer uso de una

versión previa de ADB, como la 1.0.31, sí es posible hacer una copia de seguridad del dispositivo móvil y de sus apps. La versión 1.0.31 de "adb" está disponible en la versión 20 de las "Platform Tools" [Ref.- 506].

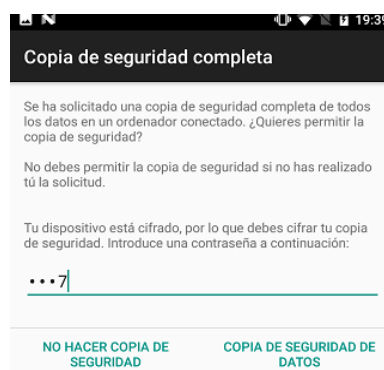
2032. Algunas apps pueden especificar en su fichero *manifest* que no permiten que se haga copia de seguridad de ellas, por lo que quedarán excluidas de este proceso.

2033. El fichero creado incluye una cabecera específica para las copias de seguridad de Android (variable según si se usa cifrado o no) y emplea el formato "tar", con ligeras modificaciones propias de Android [Ref.- 93].

```
C:\> adb backup -f Android_7x_backup.ab -all -system -shared -apk
Now unlock your device and confirm the backup operation...
```

Nota: La opción "-all" del comando "adb backup" no guarda los datos del almacenamiento compartido, por lo que aisladamente no realizará un backup completo del dispositivo. Por este motivo ha de incluirse la opción "-shared". A pesar de ello, se ha de tener precaución, pues la experiencia demuestra que ciertos datos de usuario no se mantienen tras la restauración con el comando "adb restore", como los asociados a múltiples apps.

2034. Tras ejecutar el comando "adb backup", el dispositivo móvil mostrará una pantalla de confirmación para autorizar la realización de la copia de seguridad:



2035. Cabe destacar especialmente la opción que permite establecer una contraseña para cifrar los contenidos de la copia de seguridad mediante AES 256 bits. Se recomienda siempre hacer uso de una contraseña para cifrar los contenidos de la copia de seguridad.

2036. En el caso de Android 7.x, aunque el dispositivo móvil está cifrado, no es obligatorio especificar una contraseña para proteger los contenidos de la copia de seguridad (aunque se recomienda siempre establecer una).

2037. Aunque no se especifica que sea obligatorio tener las aplicaciones cerradas a la hora de hacer un backup, sí se recomienda minimizar las apps que están abiertas, pues se han observado casos de apps que abortan su ejecución durante la operación de backup.

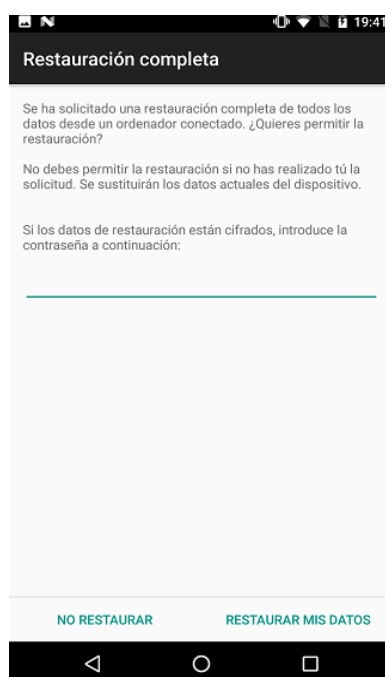
2038. El comando equivalente "adb restore" permite recuperar o restaurar una copia de seguridad realizada previamente en el dispositivo móvil:

```
C:\> adb devices
List of devices attached
0111fb17aa642917      unauthorized

<Tras habilitar en el dispositivo móvil el acceso mediante ADB proveniente del
ordenador, se podrá establecer la conexión y enviar el comando "adb restore">

C:\> adb devices
List of devices attached
0111fb17aa642917      device
C:\> adb restore Android_7x_backup.ab
Now unlock your device and confirm the restore operation.
```

2039. En el caso de que la copia de seguridad esté cifrada, el proceso muestra en la pantalla del dispositivo móvil el campo para especificar la contraseña que fue utilizada a la hora de realizar la copia de seguridad:

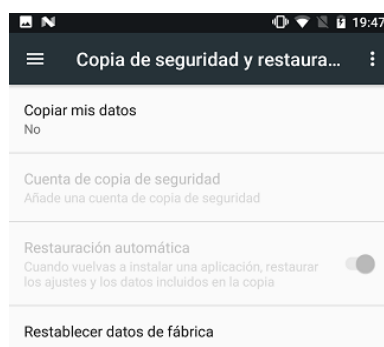


2040. En el caso de que la restauración mediante el comando "adb restore" deba realizarse partiendo de un dispositivo móvil que se acaba de restaurar a fábrica, será necesario habilitar la ejecución del comando ADB siguiendo los pasos detallados en el apartado "13.3. Opciones de desarrollo y depuración USB".

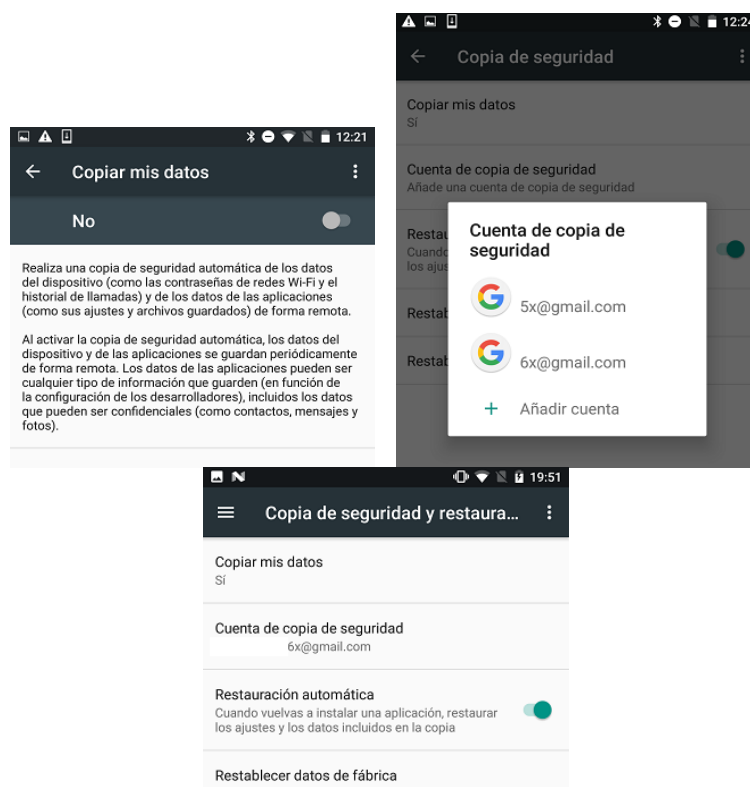
21.2. COPIA DE SEGURIDAD EN LOS SERVIDORES DE GOOGLE

2041. El servicio de copia de seguridad de datos de Google (denominado *Android Backup Service*) permite almacenar información personal y datos del usuario en los servidores de Google, como los favoritos de navegación web, palabras añadidas al diccionario, una lista de las apps instaladas, parámetros de configuración y los datos utilizados por las apps de terceros, las contraseñas de las redes Wi-Fi, la mayoría de parámetros de configuración del dispositivo móvil y de los servicios de Google, etc., siendo posible restaurar los datos posteriormente si fuese necesario reemplazar el dispositivo móvil o reinstalar una app [Ref.- 104].

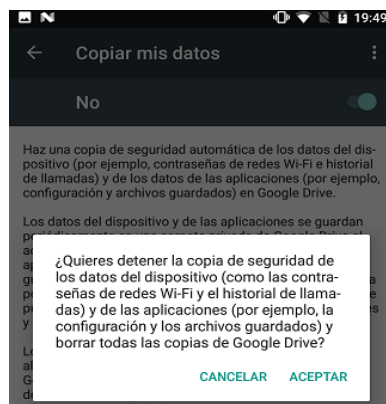
2042. Durante el proceso de instalación y configuración inicial de Android y vinculación del dispositivo móvil (ver apartado "23. Apéndice A: Proceso de instalación y configuración inicial") a una cuenta de usuario de Google (y/o de creación de la cuenta) se solicita confirmación por parte del usuario para activar el servicio de copia de seguridad de datos.
2043. En el caso de dispositivos móviles Android multi-usuario, únicamente el usuario propietario (o usuario principal) puede gestionar las opciones de copia de seguridad y restauración.
2044. El servicio de copia de seguridad de datos en los servidores de Google puede ser activado o desactivado posteriormente a través del menú "Ajustes [Personal] - Copia de seguridad", y en concreto, mediante la opción "Copiar mis datos" (siendo necesario disponer de una cuenta de usuario de Google vinculada al dispositivo móvil).



2045. En el caso en que existan varias cuentas de usuario de Google en el dispositivo móvil, será necesario elegir la cuenta de la cual se realizará la copia de seguridad (no es posible realizarla de más de una cuenta). Para seleccionarla, se debe pulsar sobre "Cuenta de copia de seguridad":



2046. Al desactivar la opción, no se llevarán a cabo más copias de seguridad de los datos en la cuenta de usuario de Google asociada, y las copias de seguridad existentes serán (supuestamente) eliminadas de los servidores de Google.



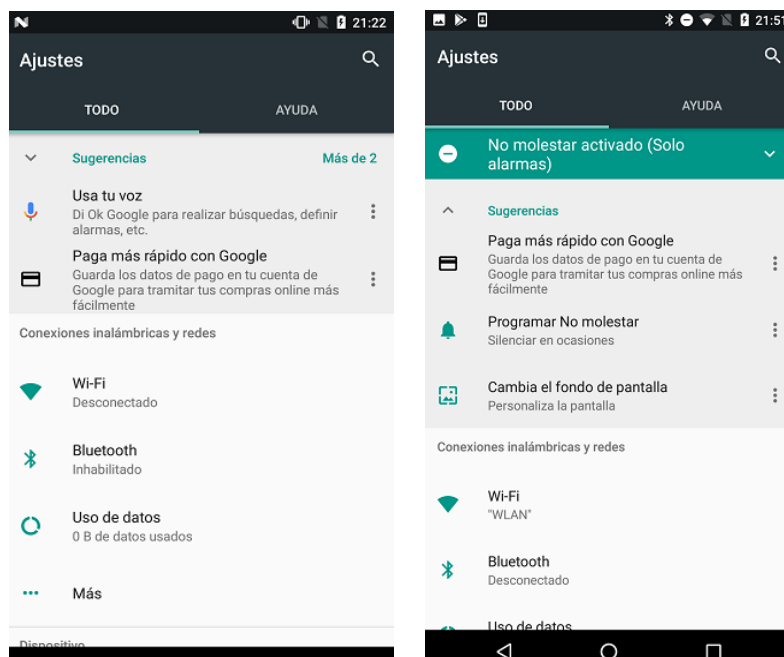
2047. Si se reinstala una app de terceros (desde Google Play), el servicio puede almacenar y restaurar la configuración y los datos asociados a la app (algunas apps pueden no permitir la realización de copias de seguridad y la restauración de sus datos).
2048. La opción "Restauración automática" permite restaurar la configuración y los datos asociados a una app al instalarla (o reinstalarla) en el dispositivo móvil desde la copia de seguridad disponible en los servidores de Google. Si la app fue instalada y usada en el mismo, o en otro dispositivo móvil, en el que se disponía del servicio de copia de seguridad de datos activo con la misma cuenta de usuario de Google, toda la información de la app será recuperada.
2049. En caso de que el usuario reemplace, sustituya o renueve su dispositivo móvil, este servicio ofrece la posibilidad de restaurar los datos desde la copia de seguridad de Google en el nuevo dispositivo móvil la primera vez que se accede a Google con la cuenta de usuario asociada a este servicio.
2050. Los datos que serán restaurados incluyen, entre otros, los ajustes de calendario y Gmail, los datos de redes Wi-Fi (incluyendo las credenciales y contraseñas de acceso), la configuración de la pantalla de inicio, los ajustes de idioma e introducción de texto, la lista de apps instaladas desde Google Play, junto a sus datos y ajustes de configuración, etc. [Ref.- 104].
2051. La opción "Restablecer ajustes de red" eliminará cualquier configuración que se haya realizado a nivel de comunicaciones en el dispositivo (incluyendo las contraseñas de las redes Wi-Fi, las redes añadidas manualmente, los emparejamientos Bluetooth, etc.).
2052. El usuario u organización propietaria del dispositivo móvil Android debe evaluar en detalle el nivel de confianza depositado en Google para emplear este servicio y almacenar todos los datos personales y confidenciales de los usuarios en el mismo.

22. INTERFAZ DE USUARIO DE ANDROID 7.X (NOUGAT)

2053. El interfaz de usuario de Android 6.x no ofrecía grandes diferencias respecto a su antecesor, Android 5.x. Sin embargo, Android 7.x introduce diversas novedades respecto a la versión anterior.

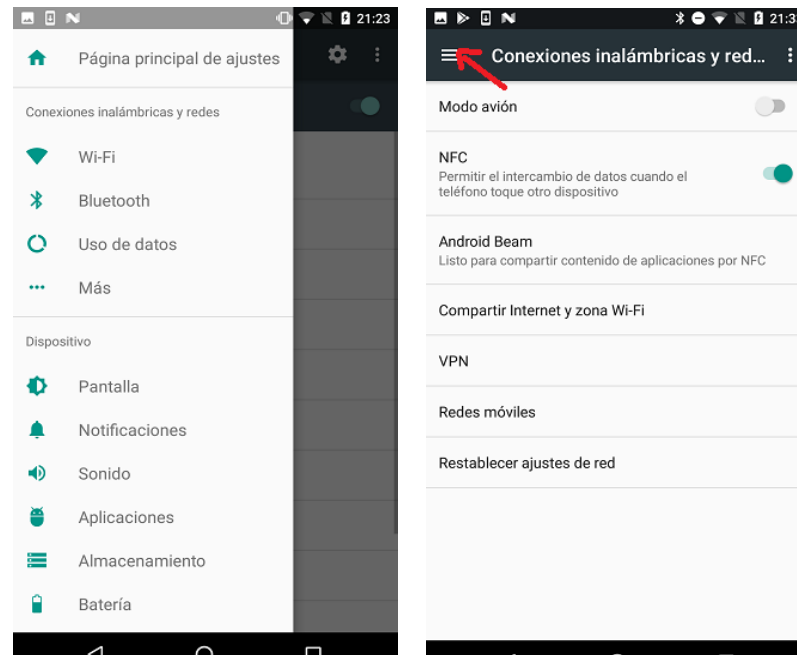
22.1. PANTALLA DE AJUSTES

2054. Se mantiene la pantalla principal (Home), pero se ha rediseñado el menú de "Ajustes" (un elemento clave en la configuración de seguridad del dispositivo móvil) y el nuevo área de notificaciones en la parte superior de la sección de "Ajustes", que pasan a tener esta apariencia:



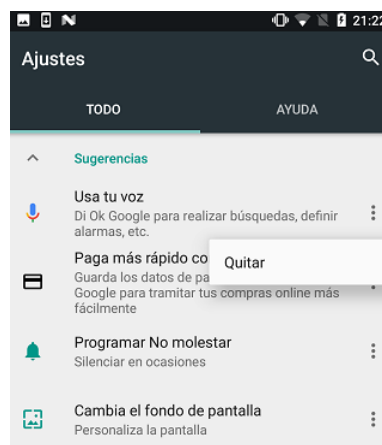
2055. El menú principal de ajustes ofrece dos cambios principales:

- Cuando se entra en la pantalla principal de "Ajustes", se presentan mensajes informativos relevantes correspondientes a las distintas categorías de configuración en la parte superior. Estos mensajes, al desplegarse, proporcionan información adicional y permiten ejecutar acciones desde los mismos. Por ejemplo, al acceder al menú de "Ajustes" estando el dispositivo en modo "No molestar", se mostrará un mensaje detallado y se presentará la opción de desactivarlo (imagen superior derecha).
- Presenta un menú compacto en la parte izquierda que contiene las categorías principales de configuración. Cuando se accede a un determinado submenú dentro de los ajustes, este menú compacto se oculta, pero se puede volver a presentar pulsando el botón formado por tres rayas horizontales del lateral superior izquierdo de la pantalla, permitiendo cambiar a otra categoría de ajuste rápidamente, sin necesidad de utilizar el botón de retroceso reiteradamente.



2056. Los mensajes informativos pueden ser eliminados del menú con un gesto, deslizando el mensaje hacia uno de los laterales de la pantalla.

2057. Adicionalmente, se presenta un apartado "Sugerencias" en la parte superior, que invita al usuario a probar nuevas funcionalidades proporcionadas por Android 7.x. Estas sugerencias se pueden eliminar desde el botón "..." situado a su derecha.

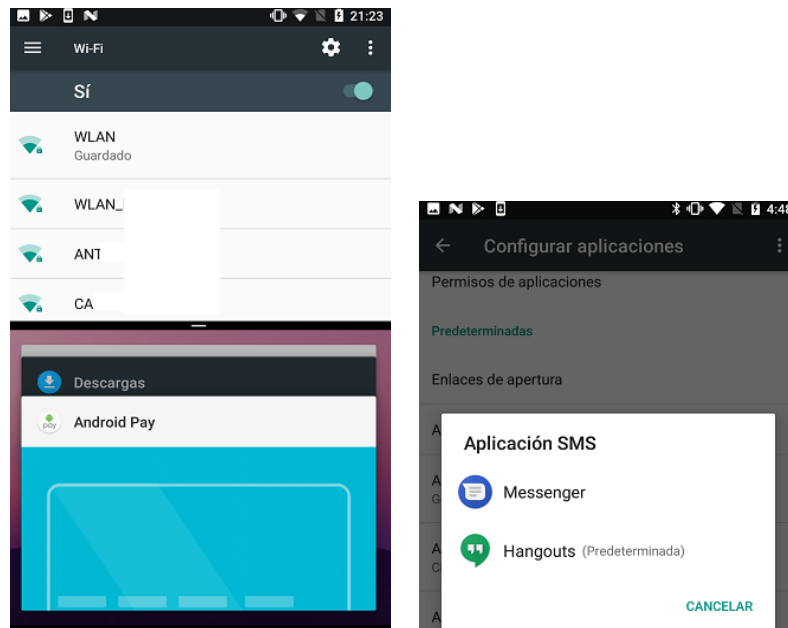


2058. La ventana principal de ajustes proporciona además la configuración relevante bajo el título del ajuste concreto de las diferentes secciones, lo que permite visualizar rápidamente la información importante y detectar ajustes erróneos o activos pero no necesarios (entre otros, la conexión involuntaria a una red Wi-Fi, volumen de datos usados en el período actual, el uso de la memoria, etc.).

22.2. SPLIT-SCREEN

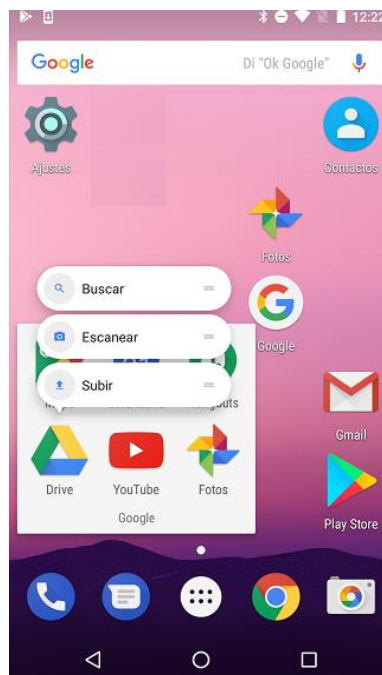
2059. Android 7.x (Nougat) implementa el modo "*Split-screen*" (pantalla dividida), que ofrece al usuario un entorno multiventana (tanto en vertical como en horizontal) para interactuar simultáneamente con dos apps.

2060. En el modo multiventana horizontal es posible establecer un tamaño diferente para cada subpantalla, mientras que en modo vertical ambas subpantallas tendrán el 50% del tamaño total.
2061. Para dividir la pantalla, se requiere realizar una pulsación larga sobre el botón inferior derecho (que proporciona acceso a las apps recientes) estando en primer plano una de ellas. Esta app ocupará la subpantalla superior (la izquierda si se está en modo horizontal), y el usuario podrá elegir la app que ocupará la otra división de entre la lista de apps que el sistema presentará de forma automática. Es importante reseñar que no todas las apps soportan trabajar con ventanas divididas.



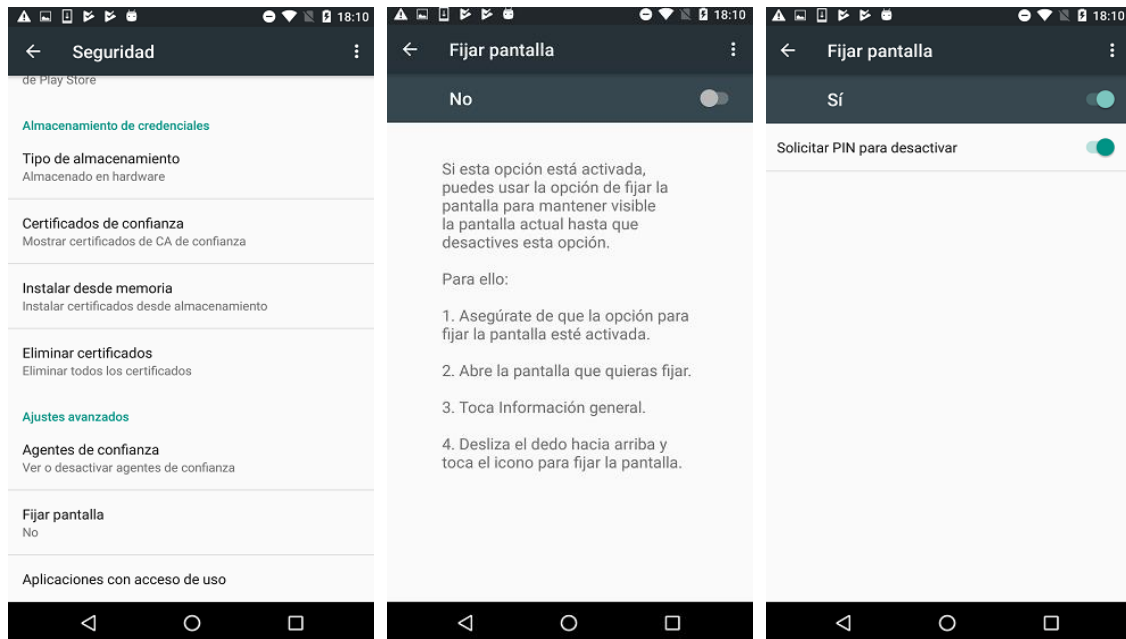
2062. También se puede acceder al modo multiventana pulsando una vez el botón de aplicaciones recientes, seleccionando mediante una pulsación larga la app que se desee fijar como principal y arrastrándola hasta el cuadro superior derecho que ofrece esta acción.
2063. En el modo multiventana, el botón de apps recientes se transforma en un botón formado por dos rectángulos iguales, uno sobre el otro.
2064. La app que ocupa la segunda pantalla puede sustituirse fácilmente, sin más que pulsar el botón de pantalla dividida y deslizarla hacia un extremo.
2065. Para salir del modo multiventana, basta con deslizar el dedo desde el margen de la app que se desea mantener en primer plano y arrastrarlo hasta el extremo opuesto.
2066. Android 7.x también implementa la funcionalidad "*Quick Switching*" para cambiar entre las dos apps usadas más recientemente, que se activa mediante una doble pulsación sobre el botón de apps recientes.
2067. Adicionalmente, la ventana secundaria de un entorno multiventana también permite la función "*Quick Switching*", por lo que es posible interactuar de forma sencilla con hasta tres apps.

2068. Al pulsar sobre el nuevo botón multiventana, se ofrece el menú "Borrar todo", que limpia la lista de apps ejecutadas en la ventana secundaria.
2069. Si se pulsa el botón Home estando en modo multiventana, todas las apps desaparecerán de la pantalla, pero el modo sigue estando activo, lo cual se muestra porque la barra de estado cambiará su color por el de la app de la subpantalla principal. Para regresar a la pantalla dividida, se pulsará el botón inferior derecho de la pantalla.
2070. Android 7.1.1 introduce accesos directos específicos para cada app en la pantalla principal (o de inicio), los cuales se despliegan al realizar una pulsación larga sobre el icono correspondiente a la app y permiten acceder a parte de la funcionalidad concreta de dicha app (según cada app) sin tener que lanzarla primero:

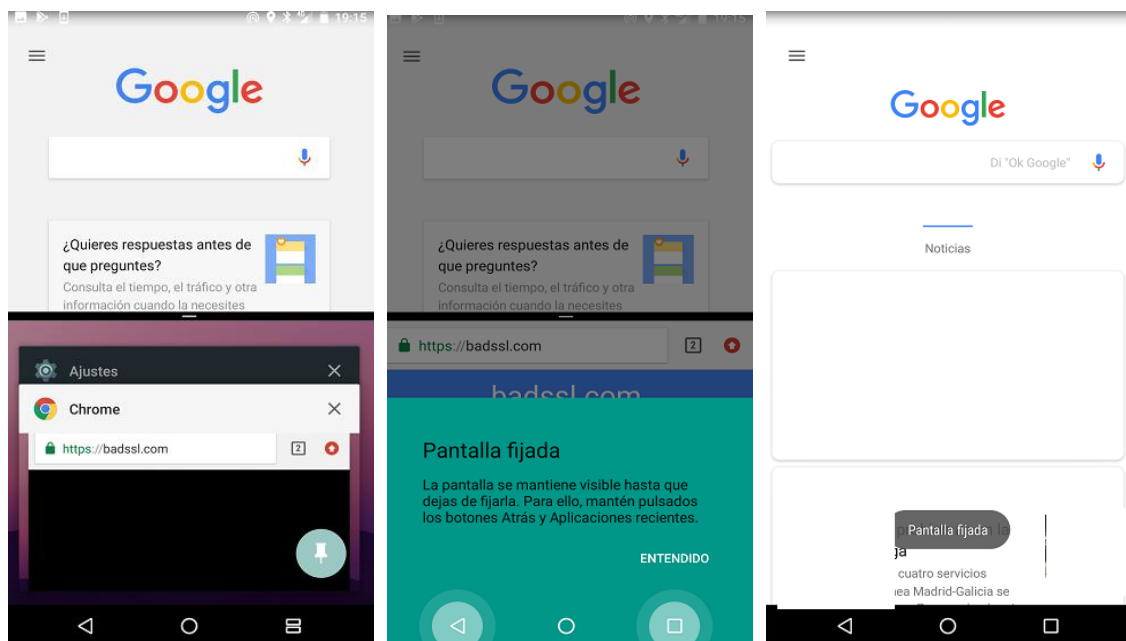


22.3. SCREEN PINNING

2071. Una de las características que introdujo Android 5.x es el denominado "*screen pinning*" (o fijación de pantalla) [Ref.- 515], que permite que el usuario pueda dejar en primer plano una aplicación para que pueda ser utilizada, y sin que sea posible cambiar a otra app distinta.
2072. Las capacidades de *screen pinning* restringen que el usuario pueda salir de la actividad o app mostrada actualmente (o que su ejecución sea interrumpida por las notificaciones), funcionalidad muy conveniente en entornos educativos o quioscos con un único propósito.
2073. Para hacer uso de esta funcionalidad, es necesario activarla desde el menú "Ajustes - [Personal] Seguridad - [Avanzado] Fijar pantalla". Se recomienda marcar la opción "Solicitar PIN para desactivar" para que, una vez se salga de la pantalla o app que se había fijado, el dispositivo móvil muestre la pantalla de bloqueo y sea necesario introducir el código de acceso para desbloquear y acceder al dispositivo.



2074. Para fijar una pantalla o app concreta, se ha de pulsar el botón de actividades o apps en ejecución, también denominado "Recientes" (correspondiente a un icono cuadrado situado en la parte inferior derecha de la pantalla) y arrastrar la barra de título de la app a fijar. Aparecerá un símbolo de *pin* en un círculo azul que, al tocarse, fijará la pantalla o app seleccionada.



2075. Para dejar de fijar una pantalla o app concreta, si esta funcionalidad fue activada manualmente por el usuario, éste deberá presionar simultáneamente los botones de "retroceso" (icono con un triángulo o flecha hacia la izquierda situado en la parte inferior izquierda de la pantalla) y (de nuevo) el botón de "Recientes" (botón con un icono cuadrado), del dispositivo móvil [Ref.- 236].



2076. El usuario no podrá acceder a la pantalla principal, ejecutar otras apps, visualizar las apps recientes o recibir notificaciones en este modo hasta que se finalice la ejecución de la app sobre la que se ha activado el *screen pinning*, independientemente de si se ha activado manualmente por parte del usuario (tal como se ha descrito previamente) o desde el código en ejecución de la propia app¹²⁵ (que debe ser una app con capacidades propietarias del dispositivo¹²⁶ para poder hacer uso de *screen pinning*; en caso contrario se solicitará autorización por parte del usuario antes de proceder a fijar la pantalla).
2077. Se recomienda también hacer uso de esta funcionalidad si, por cualquier razón, se ha de prestar temporalmente el dispositivo móvil a un tercero, pero sólo se desea que acceda a una aplicación concreta y no que disponga de capacidad para ejecutar otras aplicaciones.

22.4. CAPTURA DE LA PANTALLA EN ANDROID

2078. En versiones previas de Android hasta la versión 4.x no existía ninguna aplicación nativa para capturar la pantalla del dispositivo móvil (*screenshot*), y era necesario emplear aplicaciones de terceros que, en su mayor parte, requerían efectuar el proceso de *rooteado* del dispositivo móvil (dependiendo del modelo de terminal).
2079. Sin embargo, también era posible realizar capturas de la pantalla del dispositivo móvil Android mediante un ordenador a través del puerto USB

¹²⁵ Mediante una llamada al método "startLockTask()" o "setLockTaskPackages()".

¹²⁶ Una app con capacidades propietarias del dispositivo es una app administradora del dispositivo que no puede ser desactivada por el usuario, o desinstalada, una vez ha sido activada por el administrador en un entorno de gestión empresarial: [https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isDeviceOwnerApp\(java.lang.String\)](https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#isDeviceOwnerApp(java.lang.String)).

haciendo uso de las capacidades de depuración del kit de desarrollo de software (SDK, *Software Development Kit*) de Android.

2080. Desde Android 4.0, Ice Cream Sandwich (ICS), es posible obtener una captura de la pantalla del dispositivo móvil manteniendo pulsados el botón de encendido/apagado y el botón de bajar el volumen simultáneamente, hasta que se oiga un sonido de confirmación, o se reciba una confirmación mediante una animación visual en pantalla.
2081. Adicionalmente se puede recibir una notificación asociada a dicha captura en el centro de notificaciones disponible en la parte superior.
2082. La captura de pantalla será almacenada en la galería de fotos (o fotografías), dentro de la carpeta "Screenshots", y está disponible a través de USB mediante la carpeta "Pictures/Screenshots" de la memoria interna del dispositivo móvil o de la tarjeta externa de almacenamiento.
2083. Adicionalmente sigue siendo posible realizar una captura de la pantalla del dispositivo mediante ADB (Android Debug Bridge)¹²⁷, empleando los siguientes comandos:

```
C:\> adb shell screencap -p /sdcard/screen.png
C:\> adb pull /sdcard/screen.png
C:\> adb shell rm /sdcard/screen.png
```

2084. Por otro lado, desde Android 4.4, KitKat (nivel de API 19), es posible realizar una grabación en vídeo (en formato MPEG-4, sin audio) de la pantalla del dispositivo móvil a través de ADB. El siguiente comando permite realizar una grabación a 8 Mbps de resolución (en lugar de los 4 Mbps empleados por defecto), durante 45 segundos como máximo (en lugar de emplear la duración por defecto y máxima de 3 minutos, 180 segundos), y se almacenará el vídeo en la tarjeta SD del dispositivo móvil:

```
C:\> adb shell screenrecord --bit-rate 8000000 --time-limit 45 /sdcard/kitkat-
video.mp4 (pulsar ctrl+c para parar la grabación antes de tiempo)
```

2085. Se dispone de más información sobre las diferentes opciones de grabación de la pantalla de Android en el sitio web para desarrolladores de Android y, en concreto, de ADB¹²⁸.
2086. Si una app muestra vídeos o contenidos protegidos que no desea que sean grabados a través de esta funcionalidad, puede hacer uso del método "SurfaceView.setSecure()" para marcar el contenido como protegido.

¹²⁷ <http://developer.android.com/tools/help/adb.html>

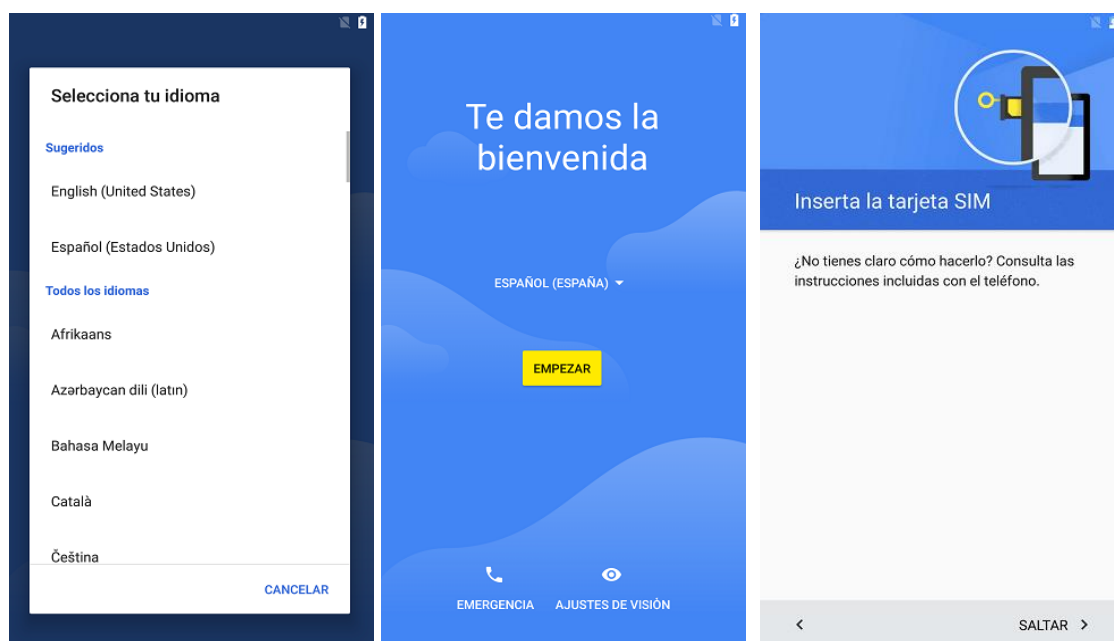
¹²⁸ <http://developer.android.com/tools/help/adb.html#screenrecord>

23. APÉNDICE A: PROCESO DE INSTALACIÓN Y CONFIGURACIÓN INICIAL

2087. El presente apartado detalla el proceso de instalación y configuración inicial de Android 7.x al encender por primera vez el dispositivo móvil tras su adquisición (o tras restablecer los ajustes de fábrica). En función de la versión concreta de Android 7.x de la que se parta, las pantallas presentadas pueden diferir en formato y/o contenido, pero las opciones de configuración serán similares. Concretamente, se ilustra la instalación de Android 7.1.1 y su posterior actualización a Android 7.1.2.

2088. Al arrancar el dispositivo móvil, se muestra una pantalla de bienvenida, "Welcome" ("Te damos la bienvenida"), donde se debe seleccionar el idioma con el que se desea configurar el dispositivo móvil y que también se empleará durante el proceso de instalación (por ejemplo, "Español (España)"). Tras seleccionar el idioma, se pulsará en el botón amarillo "Empezar".

2089. Si el proceso de instalación se aborta en cualquier momento y el terminal se reinicia, partes de la configuración son almacenadas en la partición de usuario y se tomarán esos valores como punto de partida de la configuración. Concretamente, el método de acceso, una vez configurado, permanecerá activo aunque no se haya completado la instalación.

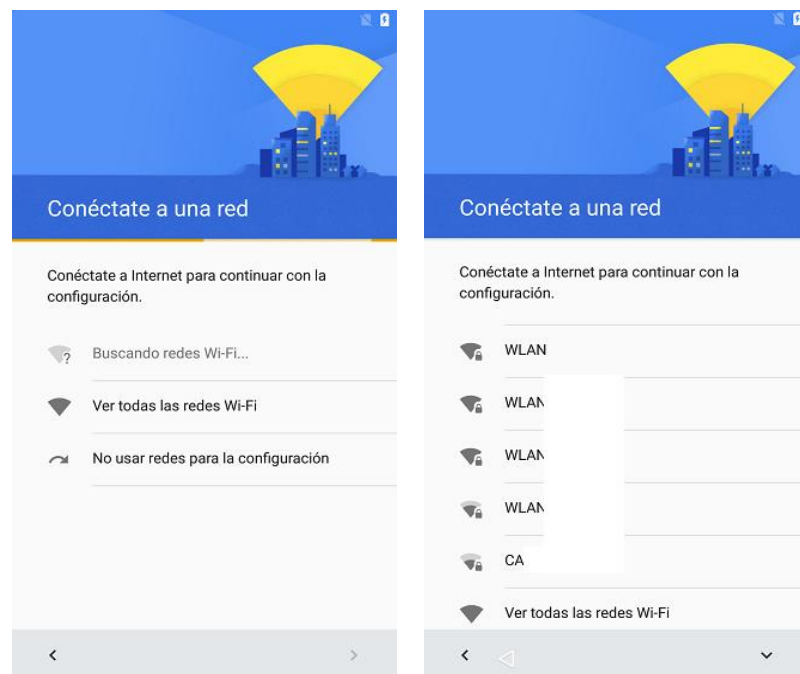


2090. Seguidamente, se muestra la pantalla que sugiere al usuario que inserte la tarjeta SIM siguiendo las instrucciones que acompañan al dispositivo móvil, "Inserta una tarjeta SIM", proceso opcional que puede ser evitado mediante el botón "Saltar" si no se desea hacer uso de las redes de telefonía móvil (2/3/4G) en este momento.

2091. El proceso de instalación ofrece la posibilidad de restaurar una copia de seguridad previa o de volcar los datos correspondientes a otro dispositivo. El presente apartado se centra en la instalación desde cero, por lo que se deberá seleccionar la opción "Configurar como nuevo":

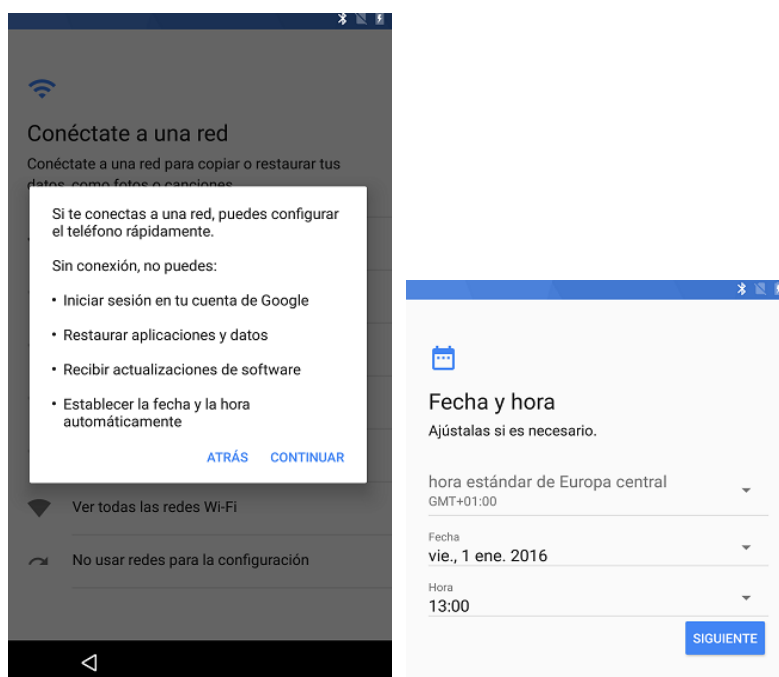


2092. Para configurar algunas de las opciones y ajustes de configuración del dispositivo móvil es necesario disponer de conectividad a Internet mediante la selección de una red Wi-Fi, "Selecciona una red Wi-Fi". Este paso también es opcional, y puede omitirse mediante la opción "No usar redes para la configuración". En caso contrario, la opción "Ver todas las redes Wi-Fi" iniciará los escaneos para detectar las redes al alcance del dispositivo móvil:

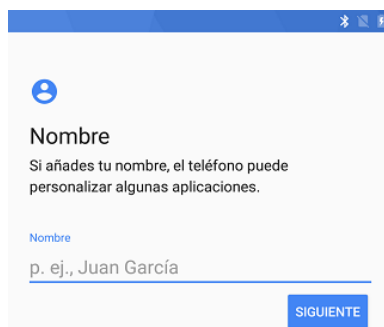


2093. Si se configura la conexión a una red Wi-Fi, se solicitará al usuario la contraseña de dicha red.
2094. En la pantalla de configuración de la red Wi-Fi se muestran las cinco redes Wi-Fi cuya intensidad de señal sea superior desde la ubicación actual del dispositivo móvil; si se desea utilizar una red Wi-Fi que no esté en dicha lista, pero que sea visible, se deberá seleccionar la opción "Ver todas las redes Wi-Fi". En ese caso, se mostrará una pantalla con todas las redes Wi-Fi visibles por el dispositivo móvil, que se actualizará dinámicamente en función de cómo varíe la recepción e intensidad de la señal de cada una de ellas. Además, si se utiliza el botón "+ Añadir nueva red", se podrán añadir redes Wi-Fi ocultas que hagan uso de diferentes mecanismos de seguridad: Abierta, WEP, WPA/WPA2 PSK o 802.1x EAP.

2095. Si se introduce incorrectamente la contraseña de la red Wi-Fi, será necesario pulsar sobre dicha red Wi-Fi hasta que aparezca el menú correspondiente, para poder seleccionar la opción "Olvidar". En tal caso, habrá que retroceder al paso anterior para que las redes Wi-Fi sean redescubiertas y volver a seleccionar la red deseada.
2096. Si no se desea utilizar una red Wi-Fi para el proceso de instalación inicial, es posible hacer uso de la opción "No usar redes para la configuración", en cuyo caso se mostrará un mensaje de aviso que informará al usuario de que, si se omite la conexión a una red Wi-Fi, el dispositivo móvil no podrá realizar descargas ni llevar a cabo las actualizaciones iniciales de software. Para continuar sin conexión, se pulsará la opción "Continuar".



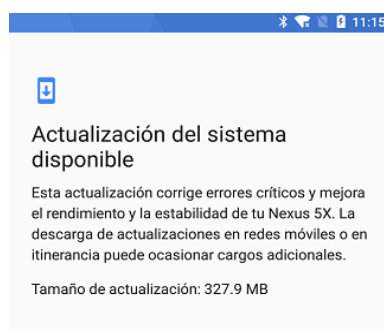
2097. Si no se ha configurado la conexión a una red Wi-Fi:
- Se presentará la pantalla de configuración de "Fecha y hora", que parte por defecto desde el año 2016, en lugar de 1970 como ocurría hasta Android 6.x.
 - Se invitará al usuario a añadir un nombre para la personalización de algunas aplicaciones. Este proceso puede omitirse sin más que dejar los campos vacíos y seleccionar la opción "Siguiente".



2098. Si se establece la conexión con una red Wi-Fi, el dispositivo móvil comprueba la disponibilidad de conectividad hacia Internet. Inicialmente

establece conexiones HTTP con "www.google.com", "android.clients.google.com", servidores del dominio "gpsnextra.net" y "clients3.google.com", y conexiones HTTPS con "www.gstatic.com".

2099. Si se introduce la configuración de una red Wi-Fi, nada más validar la existencia de una conexión hacia Internet, el dispositivo móvil muestra una pantalla que indica que se están "Buscando actualizaciones" (del sistema operativo Android). En caso de existir actualizaciones disponibles, el sistema ofrecerá información al respecto:



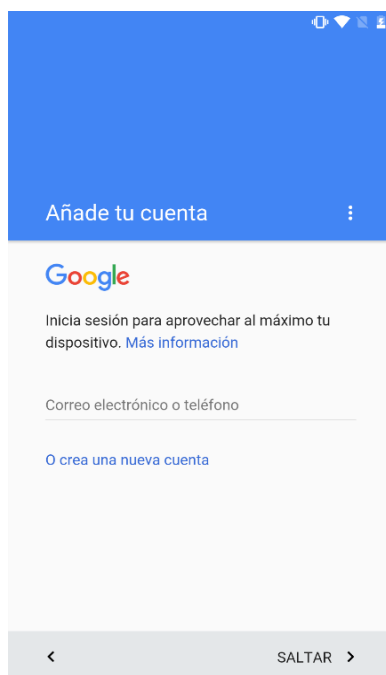
2100. La versión 5.x de Android ofrecía el menú "Tocar y listo", por el cual era posible copiar la configuración de otro dispositivo Android mediante NFC, pero esta opción no está disponible en Android 7.x.
2101. Por el contrario, se presenta la pantalla "¿Tienes otro dispositivo?" que permitiría copiar las cuentas, aplicaciones y datos desde otro dispositivo móvil. El alcance de esta guía se centra en llevar a cabo la instalación inicial completa, desde cero, por lo que se seleccionará la opción "No, gracias" (o "Configurar como nuevo" en algunas versiones de Android 7.x):



2102. Aunque en el paso anterior se opte por seleccionar la opción "No, gracias", el dispositivo móvil intentará efectuar conexiones con los servidores de Google y, si falla la conexión de red, el proceso de instalación

mostrará el mensaje "Error al iniciar sesión" (concretamente, con "https://accounts.google.com").

2103. Debe tenerse en cuenta que, tanto si se omite el proceso de conexión a una red Wi-Fi, como si se establece una conexión con una red Wi-Fi (incluso aunque posteriormente se interrumpa el proceso de configuración inicial), es posible que el interfaz Wi-Fi sea habilitado y permanezca en ese estado tras el proceso de instalación y configuración inicial de Android 7.x. Se recomienda por tanto deshabilitar el interfaz Wi-Fi lo antes posible y proceder a su configuración siguiendo las recomendaciones del apartado "16. Comunicaciones Wi-Fi".
2104. Algunas versiones de Android 7.x activan el interfaz Bluetooth desde que se inicia el proceso de instalación. Se recomienda comprobar el estado del interfaz Bluetooth tan pronto sea posible a través de "Ajustes - Conexiones inalámbricas y redes - Bluetooth" y desactivarlo en caso de estar activado, para proceder a su posterior configuración siguiendo las recomendaciones del apartado "15. Comunicaciones Bluetooth".
2105. En el caso de seleccionar una red Wi-Fi, establecer una conexión con la misma y obtener conectividad a Internet, se verifica la existencia de la conexión y se ofrece al usuario la posibilidad de configurar su cuenta de usuario de Google en el dispositivo móvil:

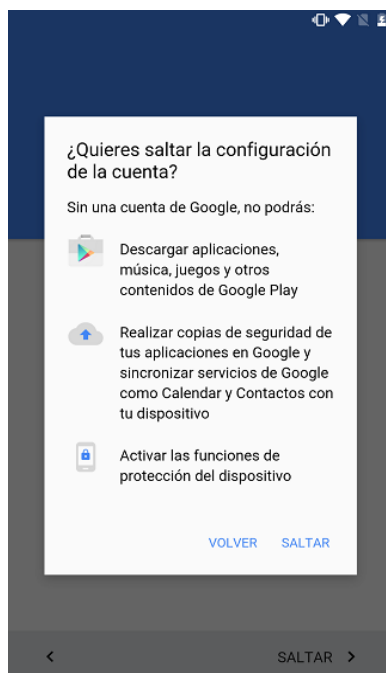


2106. Si el usuario no dispone de una cuenta de usuario de Google, se le ofrece la opción de integrar los servicios de Google en el dispositivo móvil mediante la creación de una cuenta.
2107. También se le ofrece la posibilidad, en caso de que se haya olvidado del correo electrónico asociado a la cuenta, de recuperarlo a través del número de teléfono asociado a la tarjeta SIM o de la dirección de correo electrónico de recuperación que se dio de alta al crear la cuenta de usuario de Google.

2108. Desde el punto de vista de seguridad se recomienda omitir la vinculación inicial entre el dispositivo móvil y una cuenta de usuario de Google y realizar el proceso de configuración inicial de la cuenta posteriormente, salvo que el usuario se encuentre en un entorno seguro y de confianza para poder establecer una conexión segura a una red Wi-Fi o 2/3/4G.

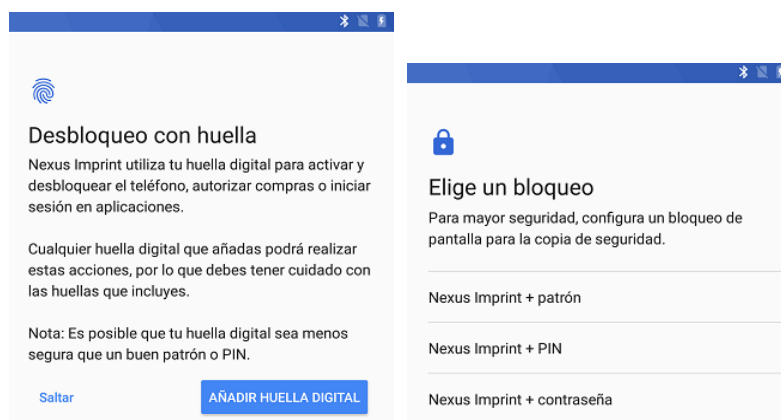
23.1. INSTALACIÓN SIN VINCULACIÓN CON CUENTA DE GOOGLE

2109. Es posible omitir el uso inicial de una cuenta de usuario de Google mediante la opción "Saltar" u "Omitir", acción que presentará una pantalla de confirmación que informa al usuario de qué implicaciones tiene no dar de alta la configuración de la cuenta de usuario de Google.



2110. La nota relativa a que, sin cuenta de usuario de Google habilitada en el dispositivo móvil, no será posible "Activar las funciones de protección del dispositivo" se refieren al denominado FRP (ver apartado "9.20. Factory Reset Protection"), pero seguirán disponibles las opciones de protección tradicionales basadas en el código de acceso.
2111. Si se opta por no crear una cuenta de Google durante la instalación, se mostrará la pantalla que permite agregar un nombre.
2112. Tras el paso anterior, se inicia la configuración de la protección de acceso al dispositivo móvil. Como novedad en Android 7.x, el proceso de instalación presenta una pantalla que parece indicar que dar de alta un método de protección de acceso implica necesariamente que se añada una huella digital mediante el botón "AÑADIR HUELLA DIGITAL" (Nexus Imprint), y seleccionar a posteriori con qué otro mecanismo de protección de respaldo se combinará la huella, pudiéndose elegir entre "Nexus Imprint + patrón", "Nexus Imprint + PIN" o "Nexus Imprint + contraseña".

2113. Esta pantalla ofrece la opción "Saltar" para el caso de que no se desee utilizar un método de protección de acceso. Si se selecciona esta opción, se informa del riesgo de uso no autorizado, y se solicita confirmación mediante la opción "Sí, SALTAR" o revisar la respuesta mediante "VOLVER".

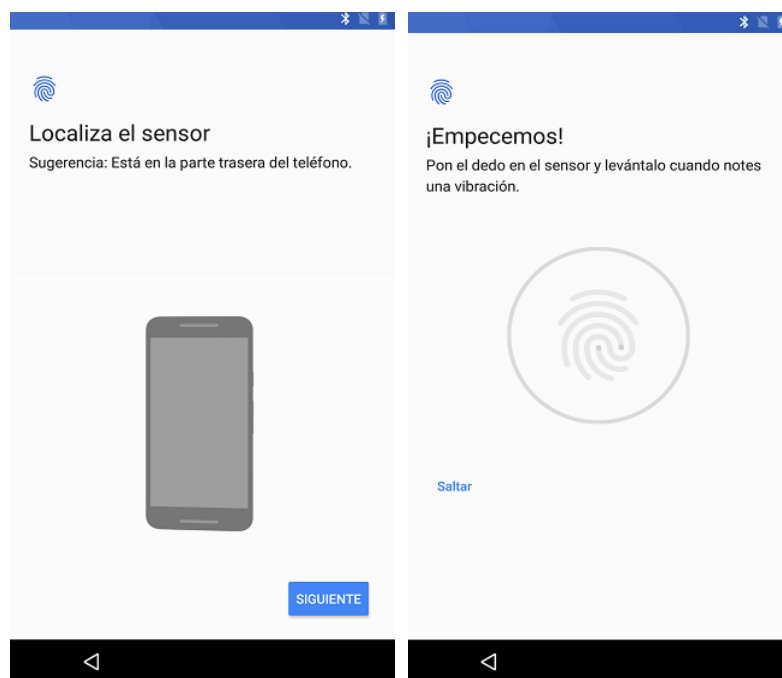


2114. Se ha comprobado que, si tras configurar el PIN, el patrón o la contraseña, lo que lleva a iniciar el proceso de adición de huella, se pulsa el botón de retroceder pantalla del dispositivo, el proceso de instalación volverá a la pantalla de configuración de la protección de acceso al dispositivo móvil y, pulsando el botón "Saltar", se obviará el proceso de configuración de una huella, pero se mantendrá el tipo de control de acceso que se hubiese ya configurado, a pesar de que se muestra un mensaje que indica lo contrario.
2115. A diferencia de otros parámetros de configuración, una vez definido un control de acceso, y aunque se pulse el botón de retroceso del dispositivo móvil para volver a pasos de configuración previos, no se vuelve a presentar el menú de activar el mecanismo de control de acceso, por lo que el primero que se haya dado de alta de forma exitosa permanecerá activo hasta que se cambien los ajustes del terminal una vez completado el proceso de instalación inicial.
2116. Tras la selección de la combinación "Nexus Imprint + <tipo de bloqueo de pantalla>", se mostrará la pantalla "Inicio seguro", que informa de que es posible proteger el dispositivo móvil de forma que el código de acceso se solicite en el arranque para poder utilizar toda la funcionalidad del dispositivo (recibir llamadas, mensajes y notificaciones, y escuchar alarmas). Si se confirma este tipo de acceso mediante "Solicitar <código> para iniciar dispositivo", se mostrará una pantalla que permite definir el código y confirmarlo otra vez más. Cuando ambos códigos coincidan, se pulsará en CONTINUAR:

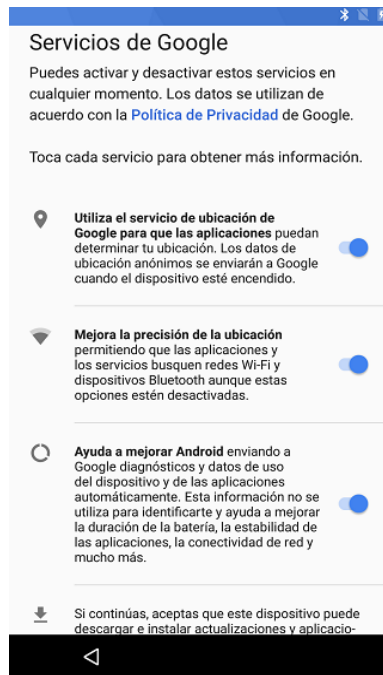


Nota: En ciertas versiones de Android 7.x, se ha constatado la existencia de un pequeño "bug" por el cual una vez seleccionado el mecanismo de protección mediante un código de acceso (patrón, PIN o contraseña), combinado con el Nexus Imprint, si antes de configurar correctamente ese mecanismo concreto se decide emplear otro y se elige "No, gracias", el dispositivo móvil mostrará pese a todo la ventana de configuración de ese método de acceso, y será necesario seleccionar en el dispositivo móvil el botón de retroceso (que aparece como un triángulo con el vértice hacia abajo en la parte inferior de la pantalla) seguido del botón de retroceso (triángulo con el vértice hacia la izquierda) para cambiar el tipo de método de acceso.

2117. Se recomienda utilizar la opción "Inicio seguro", a fin de proteger el terminal de cualquier tipo de acceso tras su reinicio y evitar el uso de una contraseña por defecto para proteger la clave maestra de cifrado.
2118. La configuración de la huella comienza con una pantalla en la que indica dónde se localiza el sensor de huella digital, ubicado en la parte posterior del Nexus 5X. Una vez localizado, se pulsa en "Siguiente" y comienza el proceso de reconocimiento de la huella, con los pasos a seguir. Será necesario colocar y levantar el dedo varias veces hasta que el terminal avise de que la huella se ha añadido:



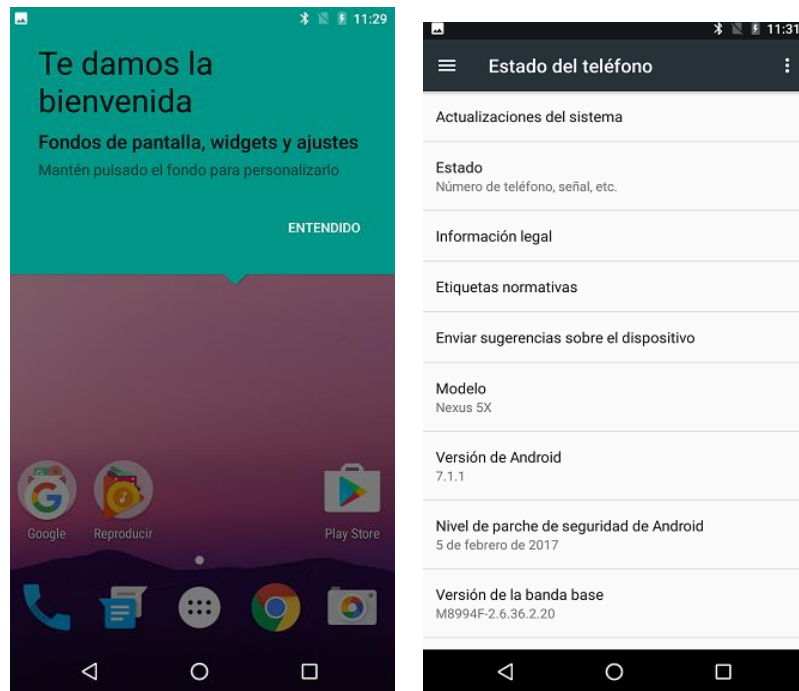
2119. Una vez aceptada la primera huella, se puede elegir añadir más huellas mediante la opción "Añadir otra", hasta un máximo de 5 huellas en total.
2120. Si transcurre un tiempo determinado entre la adición de una huella y la siguiente, aparecerá un mensaje de "Registro no completado", se interrumpirá el proceso y será preciso iniciarlo de nuevo, previa introducción del PIN, patrón o contraseña que se dio de alta como método alternativo.
2121. Una vez registradas todas las huellas que se desee, se pincha en el botón LISTO.
2122. Una vez se haya dado de alta un PIN, contraseña o patrón, si el dispositivo permanece inactivo un tiempo durante este último paso de configuración, se solicitará desbloquear el terminal mediante la opción elegida (no se incluye la opción de emplear la huella digital).
2123. Pese a que no se haya optado por vincular el dispositivo móvil con una cuenta de usuario de Google (ni se haya conectado a una red Wi-Fi), se ofrecerán ciertas opciones para activar determinadas capacidades asociadas a los servicios de ubicación de Google (equivalentes a las descritas en el apartado "23.2. Instalación vinculada a una cuenta de Google") para agilizar la localización del dispositivo por parte de las apps. Las dos opciones (habilitadas por defecto), permiten de nuevo recopilar datos de ubicación por parte de las apps y hacer uso de las capacidades de búsqueda de redes Wi-Fi para mejorar los servicios de ubicación de Google, incluso cuando el interfaz Wi-Fi esté deshabilitado:



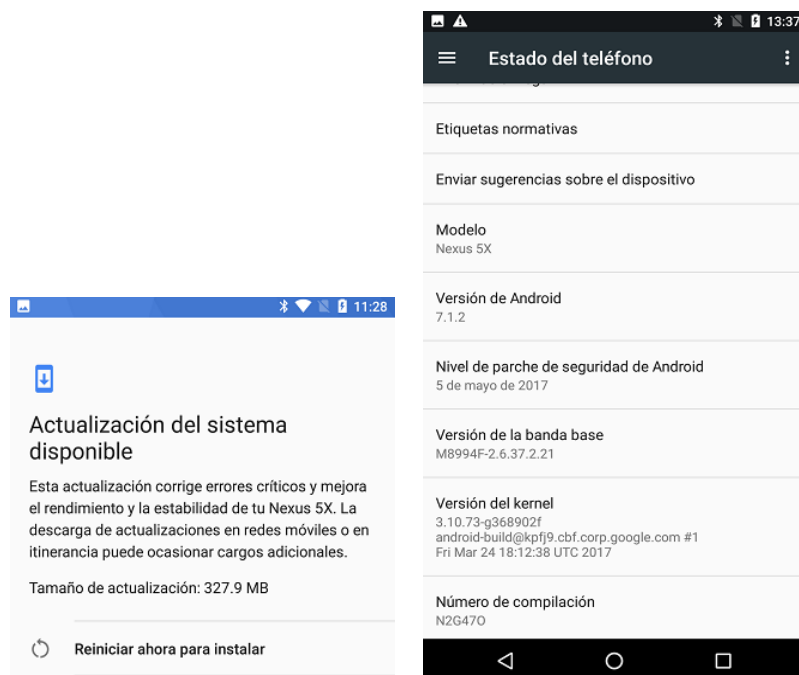
2124. Se recomienda deshabilitar ambas opciones de ubicación y habilitarlas, si se desea, posteriormente, a través del menú de "Ajustes [Personal] - Ubicación" correspondiente.
2125. En caso de deshabilitar ambas opciones, la configuración de los servicios de ubicación disponible bajo el menú "Ajustes [Personal] - Ubicación - Modo" corresponderá a la opción "Solo en dispositivo".
2126. Por último, se ofrecerá la opción de definir una cuenta de correo:



2127. El proceso de instalación concluye con la pantalla de bienvenida, tras lo cual pueden empezar a aplicarse las opciones y recomendaciones de seguridad expuestas a lo largo de la presente guía:



2128. El dispositivo mostrará una notificación indicando la existencia de una actualización. En el caso de la versión 7.1.1, se ofrecerá la actualización a la versión 7.1.2, la cual se recomienda sea aplicada desde el punto de vista de seguridad:



2129. Tras la actualización, el dispositivo móvil dispondrá de la versión 7.1.2 de Android, sobre la que se basa la mayor parte de la configuración de esta guía.

23.2. INSTALACIÓN VINCULADA A UNA CUENTA DE GOOGLE

2130. El presente apartado ilustra el proceso de instalación de Android 7.1.2 vinculando el dispositivo móvil a una cuenta de usuario de Google, que puede ser nueva o existente.

23.2.1. CREACIÓN DE UNA CUENTA DE USUARIO NUEVA

2131. Si no se dispone de una cuenta de usuario de Google y se desea que el dispositivo móvil se vincule a una, se procederá a su creación.

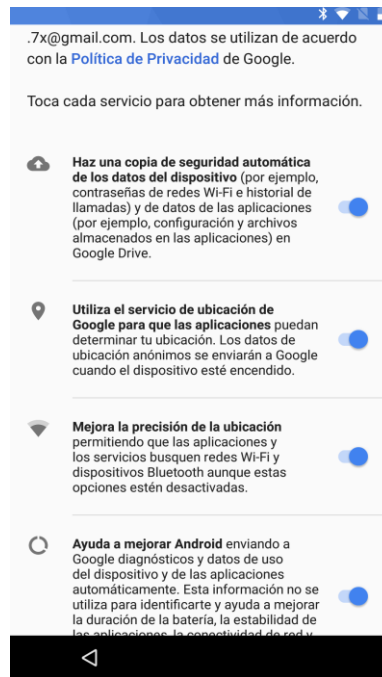
2132. Para crear una nueva cuenta de usuario de Google, se solicita el nombre y apellidos del usuario, seguido de lo que se denomina "Información Básica" y que consta de la fecha de nacimiento y el sexo del usuario (pudiéndose omitir éste). Estos campos deben rellenarse antes de proceder, si bien se permite la opción "Prefiero no decirlo" para el campo "sexo".

2133. Tras introducir la información anterior, se debe de elegir el nombre de la cuenta de usuario de Google, asociado a la cuenta de Gmail (<nombredeusuario>@gmail.com). Hay ciertos nombres de usuarios no permitidos, por ejemplo, los que contengan la palabra "android". Una vez se elige un nombre de usuario válido (que no incumpla ningún criterio no permitido, ni se encuentre ya en uso), se podrá seleccionar la contraseña asociada, y se presentará la pantalla de aceptación de los términos y condiciones del servicio.

2134. El siguiente paso propone establecer el bloqueo asociado al dispositivo mediante huella dactilar. Este proceso es similar al descrito en el apartado "23.1. Instalación sin vinculación con cuenta de Google", por lo que no se repetirá de nuevo:



2135. Tras definir el método de acceso, se iniciará sesión con la nueva cuenta de usuario de Google, empleando una conexión cifrada mediante HTTPS, y se presentan al usuario los distintos servicios de Google disponibles (todos ellos activados por defecto):

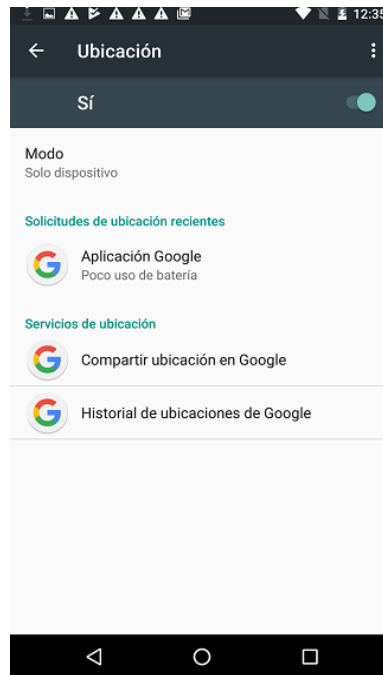


2136. Entre los servicios de Google disponibles para seleccionar, se incluye el servicio de ubicación, la ayuda para mejorar los servicios de ubicación (descritos posteriormente) y la ayuda para mejorar la experiencia del usuario con Android.

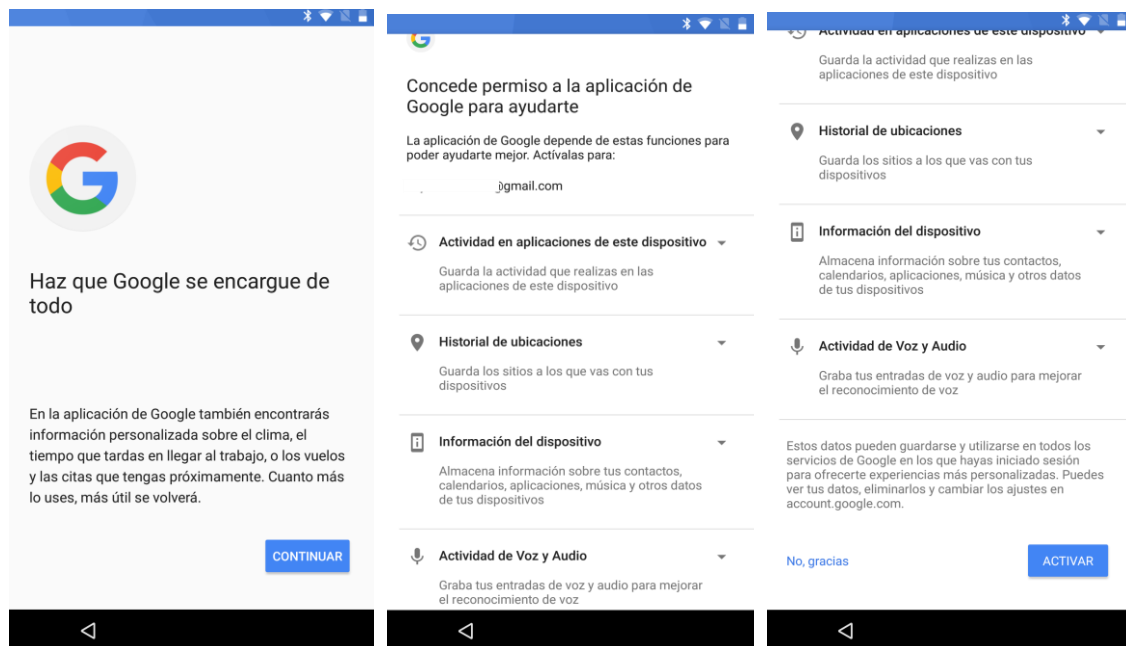
2137. Respecto a los servicios de ubicación (o localización), se dispone de dos opciones:

- Usar el servicio de ubicación de Google: es necesario para que las aplicaciones puedan determinar la ubicación del dispositivo móvil, pudiendo éstas remitir datos de ubicación anónimos a Google, aunque no estén siendo utilizadas (teóricamente del módulo GPS, de redes Wi-Fi y de las torres de telefonía móvil).
- Ayudar a mejorar los servicios de ubicación: las apps y servicios pueden identificar las redes Wi-Fi próximas (aunque el interfaz Wi-Fi del dispositivo móvil esté deshabilitado) y proporcionar dicha información a Google para mejorar la precisión de los servicios de ubicación.

2138. Se ha constatado que, aunque se hayan desmarcado todas las opciones correspondientes a los servicios de ubicación descritas previamente, el botón de activación del menú "Ajustes - [Personal] Ubicación" se quedará activo en modo "Solo dispositivo", comprobándose que los informes de ubicación de Google acceden a los servicios de ubicación (aunque en los pasos correspondientes del proceso de instalación se hubieran desmarcado las opciones de ubicación). Por tanto, se recomienda revisar de nuevo minuciosamente los ajustes de ubicación tras completar el proceso de instalación. La siguiente pantalla atestigua esta situación nada más finalizar el proceso de instalación:



2139. También aparece marcada por defecto la opción "Ayuda a mejorar Android", que será empleada para enviar datos de uso y diagnósticos a Google.
2140. Desde el punto de vista de seguridad y privacidad, se recomienda desactivar todas las opciones, salvo que se disponga de plena confianza en el tratamiento que Google hará de estos datos del usuario.
2141. La opción relativa a la copia de seguridad no aparecía en versiones anteriores de Android, y también se aconseja deshabilitarla hasta evaluar la conveniencia de este sistema de backup.
2142. En versiones anteriores de Android, el proceso de instalación ofrecía activar el servicio "Google Now" (actualmente denominado "Mi Tablón"), que proporciona información en tiempo real al usuario (este servicio también es referenciado como "*feed*"). En la versión 7.1.2, empleada en la elaboración del presente apartado, no se referencia directamente ni al concepto de "Google Now" ni "Mi Tablón", aunque la funcionalidad que se ofrece en esta pantalla es la correspondiente a esos servicios:



2143. Se recomienda no activar ninguna de estas funciones y proseguir mediante el botón "No, gracias".

2144. En caso de haberse activado dichas opciones, se iniciarán los servicios de ubicación, el envío de estadísticas a Google y el historial de ubicaciones de Google (ver apartado "12.5. Historial de ubicaciones"), y se accederá a los datos del calendario del usuario, sus búsquedas web, incluyendo el historial web y otros datos del usuario disponibles en los productos y servicios de Google (y/o de terceros):

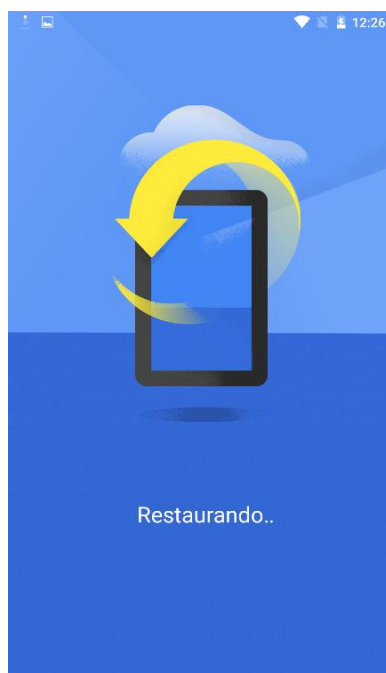
23.2.2. UTILIZACIÓN DE CUENTA DE GOOGLE EXISTENTE

2145. Si ya se dispone de cuenta de usuario de Google, las credenciales asociadas se pueden introducir en la pantalla de instalación, y aceptar las condiciones de servicio:



2146. Tras pulsar en "Acepto", los pasos de instalación son idénticos a los del apartado anterior, "23.2.1. Creación de una cuenta de usuario nueva".

2147. Tras finalizar el proceso de instalación empleando una cuenta de usuario de Google ya existente, el dispositivo recibirá de los servidores de Google las apps que estuviesen vinculadas a la cuenta como "instaladas".



24. REFERENCIAS

La siguiente tabla muestra las fuentes de información a las que se hace referencia a lo largo de la presente guía¹²⁹:

Referencia	Título, autor y ubicación
[Ref.- 1]	Android. Google. URL: http://www.android.com
[Ref.- 2]	Google Nexus 5. Google. URL: http://www.google.com/nexus/5/
[Ref.- 3]	Google Nexus 5X. Google. URL: https://www.google.com/nexus/5x/
[Ref.- 4]	Google Play. URL: https://play.google.com
[Ref.- 5]	"Exercising Our Remote Application Removal Feature". Rich Cannings. Android. June 2010. URL: http://android-developers.blogspot.com/2010/06/exercising-our-remote-application.html
[Ref.- 6]	"Actualizar las aplicaciones Android". Google. URL: https://support.google.com/googleplay/answer/113412?hl=es
[Ref.- 7]	Android Developers. Google. URL: http://developer.android.com
[Ref.- 8]	"The dark side of the new Google Play". Denis. Kaspersky Lab Expert. February 2011. URL: http://www.securelist.com/en/blog/11153/The_dark_side_of_the_new_Android_Play
[Ref.- 9]	"Android 2.3 Platform and Updated SDK Tools". December 2010. Google. URL: http://android-developers.blogspot.com/2010/12/android-23-platform-and-updated-sdk.html
[Ref.- 10]	"Android 2.3.3 Platform, New NFC Capabilities". February 2011. Google. URL: http://android-developers.blogspot.com/2011/02/android-233-platform-new-nfc.html
[Ref.- 11]	"Android Architecture Components". Google. URL: https://developer.android.com/develop/index.html
[Ref.- 12]	"SafetyNet Helper Sample". Google Play. URL: https://play.google.com/store/apps/details?id=com.scottyab.safetynet.sample
[Ref.- 13]	"Google Apps Device Policy". Google. URL: https://play.google.com/store/apps/details?id=com.google.android.apps.enterprise.dmagent URL: https://www.google.com/support/mobile/bin/answer.py?hl=en&answer=190930
[Ref.- 14]	"Google Authenticator". Google. Google Play. URL: https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2 URL: https://www.google.com/support/accounts/bin/answer.py?answer=1066447
[Ref.- 15]	"About 2-step verification". Google. URL: https://support.google.com/accounts/answer/180744
[Ref.- 16]	"Application Fundamentals". Android Developers. Google. URL: http://developer.android.com/guide/components/fundamentals.html
[Ref.- 17]	Android Developers - Direct Share. Google. URL: https://developer.android.com/samples/DirectShare/index.html
[Ref.- 18]	"Mobile Application Security". Himanshu Dwivedi, Chris Clark, David Thiel. McGraw-Hill. ISBN: 0071633561. 2010. URL: http://www.mhprofessional.com/product.php?isbn=0071633561
[Ref.- 19]	"API Levels". Android Developers. Google. URL: http://developer.android.com/guide/appendix/api-levels.html

¹²⁹ El texto "[old]" indica que se tiene constancia de que el enlace pudiera no estar operativo, o estar desactualizado, en la actualidad, pero se mantiene debido a su relevancia y por motivos históricos. En algunos casos se acompaña de una referencia actualizada, indicada por el texto "[actualizada]".

Referencia	Título, autor y ubicación
[Ref.- 20]	"Verificación en dos pasos (two-step verification)". Google. URL: https://www.google.com/landing/2step/ URL: https://www.google.com/landing/2step/features.html
[Ref.- 21]	"Android VPN Connectivity". pfSense. URL: https://doc.pfsense.org/index.php/Android_VPN_Connectivity
[Ref.- 22]	Android Studio & SDK (Software Development Kit). Google. URL: http://developer.android.com/sdk/index.html
[Ref.- 23]	Manifest Permission (Since API Level 1). Google. URL: http://developer.android.com/reference/android/Manifest.permission.html
[Ref.- 24]	"Android Security Reports". Google Report. Android security Center. URL: https://source.android.com/security/overview/reports
[Ref.- 25]	"SafetyNet: Google's tamper detection for Android". John Kozyrakis. URL: https://koz.io/inside-safetynet/
[Ref.- 26]	"Code signing in Android's security model". Nikolay Elenkov. May 2013. URL: http://nelenkov.blogspot.com.es/2013/05/code-signing-in-androids-security-model.html
[Ref.- 27]	"BouncyCastle Key Store (BKS) library". The Legion of the Bouncy Castle. URL: http://bouncycastle.org/download/bcprov-jdk16-141.jar
[Ref.- 28]	"Signing Your Applications". Android Developers. URL: http://developer.android.com/guide/publishing/app-signing.html
[Ref.- 29]	RedPhone. Open WhisperSystems. 2015. URL: https://whispersystems.org URL: https://play.google.com/store/apps/details?id=org.thoughtcrime.redphone
[Ref.- 30]	"Getting Your SMS Apps Ready for KitKat". Android Developers Blog. October 2013. URL: http://android-developers.blogspot.com.es/2013/10/getting-your-sms-apps-ready-for-kitkat.html
[Ref.- 31]	"Android 3.0 Platform Highlights". Google. 2011. URL: http://developer.android.com/sdk/android-3.0-highlights.html#enterprise
[Ref.- 32]	Signal (Private Messenger). Open Whisper Systems. URL: https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms URL: https://github.com/WhisperSystems/Signal-Android
[Ref.- 33]	TextSecure. Open Whisper Systems. 2015. URL: https://whispersystems.org URL: https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms
[Ref.- 34]	"Using KitKat verified boot". Nikolai Elenkov. May 2014. URL: https://nelenkov.blogspot.com/2014/05/using-kitkat-verified-boot.html
[Ref.- 35]	"SafetyNet Playground". Google Play. URL: https://play.google.com/store/apps/details?id=com.cigital.safetynetplayground
[Ref.- 36]	"Understanding Exchange ActiveSync Mailbox Policies". MS Exchange 2010 SP2 & SP3. Microsoft Technet. 2014. URL: https://technet.microsoft.com/en-us/library/bb123484.aspx
[Ref.- 37]	"View or Configure Exchange ActiveSync Mailbox Policy Properties" (Microsoft Exchange 2010). Microsoft Technet. 2010. URL: http://technet.microsoft.com/en-us/library/bb123994.aspx
[Ref.- 38]	"Android support for Microsoft Exchange in pure Google devices". Google. 2013. URL: http://static.googleusercontent.com/media/www.google.com/es//help/hc/images/android/MicrosoftExchangePoliciesinAndroid.pdf
[Ref.- 39]	"Issue 82: wifi - support ad hoc networking". Android Google Code. 2008. URL: https://code.google.com/p/android/issues/detail?id=82
[Ref.- 40]	"Ad-Hoc (IBSS) mode support for Android 4.2.2 / 4.3 / 4.4". Thinktube. March 2014. URL: http://www.thinktube.com/tech-en/android/wifi-ibss
[Ref.- 41]	"Why Wi-Fi Direct cannot replace Ad-hoc mode". szym. Thinktube. 2014. URL: http://www.thinktube.com/tech-en/android/wifi-direct
[Ref.- 42]	"Issue 1041: Hidden SSID Fails to Connect". Android Google Code. 2008. URL: https://code.google.com/p/android/issues/detail?id=1041
[Ref.- 43]	"How can I trust CAcert's root certificate?". CAcert Wiki. URL: http://wiki.cacert.org/ImportRootCert#Android_Phones

Referencia	Título, autor y ubicación
[Ref.- 44]	"Android code signing". Nikolay Elenkov. April 2013. URL: http://nelenkov.blogspot.com.es/2013/04/android-code-signing.html
[Ref.- 45]	"RealmB's Android Certificate Installer". RealmB. URL: http://www.realmmb.com/droidCert/
[Ref.- 46]	"Settings-Secure". Android Developers. URL: http://developer.android.com/reference/android/provider/Settings.Secure.html
[Ref.- 47]	dhcpd. Roy Marples. URL: https://roy.marples.name/projects/dhcpd
[Ref.- 48]	"Guía CCN-STIC-406: Seguridad en redes inalámbricas". CCN-CERT. URL: https://www.ccn-cert.cni.es (Guías)
[Ref.- 49]	"Description of the Wireless Client Update for Windows XP with Service Pack 2". Microsoft. 2010. URL: http://support.microsoft.com/kb/917021
[Ref.- 50]	"Guía CCN-STIC-418: Seguridad en Bluetooth". CCN-CERT. URL: https://www.ccn-cert.cni.es (Guías)
[Ref.- 51]	"ConnectivityManager". Android Developers. URL: https://developer.android.com/reference/android/net/ConnectivityManager.html
[Ref.- 52]	"Wi-Fi Direct ("P2P")". Wi-Fi Alliance. 2013. URL: http://www.wi-fi.org/discover-and-learn/wi-fi-direct
[Ref.- 53]	"Issue 3389: Full IPv6 Android support". Android Google Code. 2009. URL: https://code.google.com/p/android/issues/detail?id=3389 URL: https://code.google.com/p/android/issues/detail?id=3389#c107
[Ref.- 54]	"Issue 14013: Enable IPv6 Privacy Extensions". Android Google Code. 2011. URL: https://code.google.com/p/android/issues/detail?id=14013 URL: http://www.h-online.com/security/news/item/IPv6-Smartphones-compromise-users-privacy-1169708.html
[Ref.- 55]	"These aren't the permissions you're looking for". A. Lineberry, D. Richardson, T. Wyatt". Defcon 18. 2010. URL: https://media.defcon.org/DEF%20CON%2018/DEF%20CON%2018%20slides/DEF%20CON%2018%20Hacking%20Conference%20Presentation%20By%20-%20Lineberry%20Richardson%20and%20Wyatt%20-%20These%20Aren%27t%20the%20Permissions%20-%20Slides.m4v
[Ref.- 56]	"Support connecting to IPv6-only wireless networks". Android Google Code. June 2012. URL: https://code.google.com/p/android/issues/detail?id=32630
[Ref.- 57]	"Best Practices for Security & Privacy". Android Developers. URL: https://developer.android.com/training/best-security.html
[Ref.- 58]	"Use of IPv6 privacy extensions should be configurable ". Android Google Code. May 2012. URL: https://code.google.com/p/android/issues/detail?id=31102
[Ref.- 59]	"2 big points critics always miss about Android upgrades". Computer World. June 2015. URL: http://www.computerworld.com/article/2476391/android/2-big-points-critics-always-miss-about-android-upgrades.html
[Ref.- 60]	"Encontrar un dispositivo Android que has perdido". Google. URL: https://support.google.com/accounts/answer/3265955?hl=es
[Ref.- 61]	"12 Days of HaXmas: A year of Metasploit Android exploits". Rapid 7. January 2015. URL: https://community.rapid7.com/community/metasploit/blog/2015/01/02/2014-a-year-of-android-exploits-in-metasploit
[Ref.- 62]	"WebView addJavascriptInterface Remote Code Execution". MRW Labs. September 2013. URL: https://labs.mwrinfosecurity.com/blog/2013/09/24/webview-addjavascriptinterface-remote-code-execution/ URL: https://labs.mwrinfosecurity.com/blog/2012/04/23/adventures-with-android-webviews/
[Ref.- 63]	"Por qué se debe iniciar sesión en Chrome". Google. URL: https://support.google.com/chrome/answer/165139?hl=es URL: https://support.google.com/chrome/answer/1181035 URL: https://support.google.com/chrome/answer/2591582?hl=es
[Ref.- 64]	Android Vulnerabilities. 2015. URL: http://androidvulnerabilities.org

Referencia	Título, autor y ubicación
[Ref.- 65]	"No puedes desbloquear el dispositivo Android". Google. URL: https://support.google.com/nexus/answer/3388218?hl=es
[Ref.- 66]	"Android Security Announcements". Google Groups. 2008. URL: https://groups.google.com/group/android-security-announce
[Ref.- 67]	"Android Security Discussions". Google Groups. 2008. URL: http://groups.google.com/group/android-security-discuss
[Ref.- 68]	[old] "Android Security FAQ". Android Developers. URL: http://developer.android.com/resources/faq/security.html URL [actualizada]: https://developer.android.com/training/best-security.html
[Ref.- 69]	"Android Security Bulletins (or Updates)". Google. August 2015. URL: https://source.android.com/security/bulletin/ URL: https://groups.google.com/forum/#!forum/android-security-updates
[Ref.- 70]	"LG Security Bulletins". LG. URL: https://lgsecurity.lge.com/security_updates.html
[Ref.- 71]	"Google Play Devices". Google. URL: https://play.google.com/store/devices
[Ref.- 72]	"Google Experience Devices". DanKor. 2012. URL: http://about.me/googleexperiencedevice URL: http://googleexperiencedevice.blogspot.com
[Ref.- 73]	"Factory Images for Nexus and Pixel Devices". Google. URL: https://developers.google.com/android/images
[Ref.- 74]	"Nexus help. Check & update your Android version". Google. URL: https://support.google.com/nexus/answer/4457705
[Ref.- 75]	"History of the Nexus family". Android Authority. URL: http://www.androidauthority.com/history-nexus-smartphone-line-536352/
[Ref.- 76]	"Android WiFi-Direct Denial of Service". Core Security. January 2015. URL: http://www.coresecurity.com/advisories/android-wifi-direct-denial-service
[Ref.- 77]	"Android Screen Lock Bypass". Thomas Cannon. 2011. URL: http://thomascannon.net/blog/2011/02/android-lock-screen-bypass/
[Ref.- 78]	"Android Market Security". Thomas Cannon. 2011. URL: http://thomascannon.net/blog/2011/02/android-market-security/
[Ref.- 79]	"Certificate pinning in Android 4.2". Nikolay Elenkov. December 2012. URL: http://nelenkov.blogspot.com/2012/12/certificate-pinning-in-android-42.html
[Ref.- 80]	"A Peek Inside the GTalkService Connection". Jon Oberheide. 2010. URL: http://jon.oberheide.org/blog/2010/06/28/a-peek-inside-the-gtalkservice-connection/
[Ref.- 81]	"Remote Kill and Install on Google Android". Jon Oberheide. 2010. URL: http://jon.oberheide.org/blog/2010/06/25/remote-kill-and-install-on-google-android/
[Ref.- 82]	"Apple. Google Collect User Data". Wall Street Journal. 2011. URL: http://on.wsj.com/gDfmEV
[Ref.- 83]	"Android Map". Samy Kamkar. 2011. URL: http://samy.pl/androidmap/
[Ref.- 84]	Android Open Source Project (AOSP). URL: http://source.android.com
[Ref.- 85]	"Android Encryption". Android. URL: https://source.android.com/security/encryption/

Referencia	Título, autor y ubicación
[Ref.- 86]	"Android X.Y Platform Highlights" & "Welcome to Android 4.x" & "Android <version>". Android Developers. URL (2.2): http://developer.android.com/about/versions/android-2.2-highlights.html URL (2.3): http://developer.android.com/about/versions/android-2.3-highlights.html URL (3.0): http://developer.android.com/about/versions/android-3.0-highlights.html URL (4.0): http://developer.android.com/about/versions/android-4.0-highlights.html URL (4.1): http://developer.android.com/about/versions/jelly-bean.html#android-41 URL (4.2): http://developer.android.com/about/versions/jelly-bean.html#android-42 URL (4.3): http://developer.android.com/about/versions/jelly-bean.html#android-43 URL (4.4): https://developer.android.com/about/versions/kitkat.html URL (5.0): https://developer.android.com/about/versions/lollipop.html URL (6.0): https://developer.android.com/about/versions/marshmallow/index.html
[Ref.- 87]	"Compatibility Test Suite (CTS)". Google. URL: http://source.android.com/compatibility/cts/index.html
[Ref.- 88]	"Malware-Infected Apps Yanked From Android Market". Thread post. URL: https://threatpost.com/malware-infected-apps-yanked-android-market-030211/74985/
[Ref.- 89]	CTS (Compatibility Test Suite). AOSP. URL: http://source.android.com/compatibility/cts-intro.html
[Ref.- 90]	"Checking Device Compatibility with SafetyNet". Android Developers. URL: https://developer.android.com/training/safetynet/index.html
[Ref.- 91]	"Device Policy for Android" & "Device Policy Administration for Android" & "Google Apps Mobile Management: Overview". Google Mobile. URL: https://support.google.com/mobile/bin/answer.py?hl=en&answer=190930 URL: http://support.google.com/a/bin/answer.py?hl=en&answer=1056433 URL: https://support.google.com/a/bin/answer.py?hl=en&answer=1734200
[Ref.- 92]	"Issue 6111: Allow a hostname to be specified". Android Google Code. January 2010. URL: https://code.google.com/p/android/issues/detail?id=6111
[Ref.- 93]	"Unpacking Android backups". Nikolay Elenkov. June 2012. URL: http://nelenkov.blogspot.com.es/2012/06/unpacking-android-backups.html URL: https://github.com/nelenkov/android-backup-extractor
[Ref.- 94]	"Establecer conexión mediante zona Wi-Fi, Bluetooth o USB". Google Mobile. URL: https://support.google.com/nexus/answer/2812516?hl=es
[Ref.- 95]	"[TAD-2011-003] Vulnerability in Android: To add, or not to add (a Wi-Fi network), that is the question". Raúl Siles. Taddong. Mayo 2010. URL: http://blog.taddong.com/2011/05/vulnerability-in-android-to-add-or-not.html
[Ref.- 96]	"Issue 1215: IP settings for WiFi networks". Android Android Google Code. November 2008. URL: https://code.google.com/p/android/issues/detail?id=1215
[Ref.- 97]	"Putting Android to work for your business". Official Google Enterprise Blog. April 2011. URL: http://googleenterprise.blogspot.com/2011/04/putting-android-to-work-for-your.html
[Ref.- 98]	"SafetyNet". Google. URL: https://developer.android.com/training/safetynet/index.html
[Ref.- 99]	"Issue 11855: Support settings change in Android WIFI/USB Tether". Android Google Code. October 2010. URL: https://code.google.com/p/android/issues/detail?id=11855
[Ref.- 100]	"A Tale Of Another SOP Bypass In Android Browser < 4.4". Rafay Baloch . October 2014. URL: http://www.rafayhackingarticles.net/2014/10/a-tale-of-another-sop-bypass-in-android.html URL: https://community.rapid7.com/community/metasploit/blog/2014/11/06/metasploit-weekly-wrapup
[Ref.- 101]	"Android Fake ID bug exposes smartphones and tablets". BBC. URL: http://www.bbc.com/news/technology-28544443
[Ref.- 102]	"Package javax.net.ssl". Android Developers. URL: https://developer.android.com/reference/javax/net/ssl/SSLContext
[Ref.- 103]	"Magisk (Manager)". XDA Developers. URL: http://forum.xda-developers.com/apps/magisk/official-magisk-v7-universal-systemless-t3473445

Referencia	Título, autor y ubicación
[Ref.- 104]	"Copia de seguridad y restauración de datos del dispositivo". Google. URL: https://support.google.com/nexus/answer/2819582?hl=es
[Ref.- 105]	"The AndroidManifest.xml File: <permission>". Android Developers. URL: http://developer.android.com/guide/topics/manifest/permission-element.html#plevel
[Ref.- 106]	"SE Android". SE Linux Project. 2012. URL: https://seandroid.bitbucket.io/ URL: http://selinuxproject.org/~jmorris/lss2011_slides/caseforseandroid.pdf
[Ref.- 107]	"Configurar el control parental". Google Play. URL: https://support.google.com/googleplay/answer/1075738?hl=es
[Ref.- 108]	"Flash to Focus on PC Browsing and Mobile Apps; Adobe to More Aggressively Contribute to HTML5". Adobe. Noviembre 2011. URL: http://blogs.adobe.com/conversations/2011/11/flash-focus.html
[Ref.- 109]	"Issue 10809: Password is stored on disk in plain text". Android Google Code. August 2010. URL: https://code.google.com/p/android/issues/detail?id=10809
[Ref.- 110]	"Android Debug Bridge (ADB)". Android Developers. URL: http://developer.android.com/guide/developing/tools/adb.html
[Ref.- 111]	"Catching AuthTokens in the Wild - The Insecurity of Google's ClientLogin Protocol". Bastian Könings, Jens Nickels, and Florian Schaub. Mayo 2011. URL: http://www.uni-ulm.de/en/in/mi/staff/koenings/catching-authtokens.html URL: https://freedom-to-tinker.com/blog/dwallach/things-overheard-wifi-my-android-smartphone URL: http://android.stackexchange.com/questions/3129/what-android-syncd-data-is-encrypted
[Ref.- 112]	"Chrome for Android". Google. URL: https://developer.chrome.com/multidevice/faq
[Ref.- 113]	"Google Data Protocol - FAQ". Google Code. URL: https://code.google.com/apis/gdata/faq.html#clientlogin_expire
[Ref.- 114]	"Authentication and Authorization for Google APIs - ClientLogin for Installed Applications". Google Code. URL: https://code.google.com/apis/accounts/docs/AuthForInstalledApps.html
[Ref.- 115]	"Using OAuth 2.0 to Access Google APIs". Google. URL: https://developers.google.com/accounts/docs/OAuth2
[Ref.- 116]	"LOOK-10-007 - TapJacking". Lookout. December 2010. URL: http://blog.mylookout.com/look-10-007-tapjacking/ URL: http://blog.mylookout.com/blog/2010/12/09/android-touch-event-hijacking/
[Ref.- 117]	[old] "Android Master Key Exploit - Uncovering Android Master Key That Makes 99% of Devices Vulnerable". BlueBox. July 2013. URL: https://bluebox.com/technical/uncovering-android-master-key-that-makes-99-of-devices-vulnerable/
[Ref.- 118]	"Android Browser Same Origin Policy Bypass < 4.4 - CVE-2014-6041". Rafay Baloch . September 2014. URL: http://www.rafayhackingarticles.net/2014/08/android-browser-same-origin-policy.html URL: https://community.rapid7.com/community/metasploit/blog/2014/09/15/major-android-bug-is-a-privacy-disaster-cve-2014-6041
[Ref.- 119]	"Android WebView fixes". Adrian Ludwig. January 2015. URL: https://plus.google.com/+AdrianLudwig/posts/1md7ruEwBLF URL: https://community.rapid7.com/community/metasploit/blog/2015/01/11/google-no-longer-provides-patches-for-webview-jelly-bean-and-prior URL: https://community.rapid7.com/community/metasploit/blog/2015/01/16/weekly-metasploit-wrapup
[Ref.- 120]	"Issue 8196: Enhancement: Client Certificate Authentication in Browser". Android Google Code. May 2010. URL: https://code.google.com/p/android/issues/detail?id=8196
[Ref.- 121]	"Issue 11231: Provide support for managing CA and client certificates". Android Google Code. September 2010. URL: https://code.google.com/p/android/issues/detail?id=11231
[Ref.- 122]	"KeyChain". Android Developers. URL: http://developer.android.com/reference/android/security/KeyChain.html
[Ref.- 123]	"Browser Security Handbook, part 2". Michal Zalewski. Marzo 2011 (actualizado). URL: https://code.google.com/p/browsersec/wiki/Part2#Same-origin_policy_for_cookies

Referencia	Título, autor y ubicación
[Ref.- 124]	"Uso del ID de publicidad de Android": Google Play. URL: https://play.google.com/intl/es/about/monetization-ads/ads/ad-id/
[Ref.- 125]	"Issue 6348: Bluetooth should have the option to be constantly discoverable and not just have a set time limit". Android Google Code. January 2010. URL: https://code.google.com/p/android/issues/detail?id=6348
[Ref.- 126]	"Malicious Android Applications: Risks and Exploitation". Joany Boutet. SANS Reading Room. Marzo 2010. URL: https://www.sans.org/reading_room/whitepapers/malicious/malicious-android-applications-risks-exploitation_33578
[Ref.- 127]	"App Ops Brings Granular Permissions Control to Android 4.3". Will Verduzco. Julio 2013. URL: http://www.xda-developers.com/app-ops-brings-granular-permissions-control-to-android-4-3/ URL: http://www.androidpolice.com/2013/07/25/app-ops-android-4-3s-hidden-app-permission-manager-control-permissions-for-individual-apps/
[Ref.- 128]	"App Ops Lives On In Android 4.4, Can Now Deny Even More Permissions". Android Police. November 2013. URL: http://www.androidpolice.com/2013/11/25/new-app-app-ops-lives-on-in-android-4-4-can-be-unleashed-with-app-ops-from-color-tiger-and-enhanced-with-root/
[Ref.- 129]	"Using WebView - Security Tips". Google. 2015. URL: http://developer.android.com/training/articles/security-tips.html#WebView
[Ref.- 130]	"Android Version History". Wikipedia. URL: https://en.wikipedia.org/wiki/Android_version_history
[Ref.- 131]	"AppOps". Sylvain Galand. URL: https://play.google.com/store/apps/details?id=fr.slvn.appops&hl=en URL: https://github.com/slvn/android-appops-launcher
[Ref.- 132]	"Security". Android. URL: https://www.android.com/security-center/
[Ref.- 133]	"Security". Android Open Source. URL: https://source.android.com/security/index.html
[Ref.- 134]	"Revisiting Android disk encryption". Nikolay Elenkov. Octubre 2014. URL: http://nelenkov.blogspot.com.es/2014/10/revisiting-android-disk-encryption.html
[Ref.- 135]	"DevBytes: Android L Developer Preview". Google Developers. Julio 2014. URL: https://www.youtube.com/watch?v=s7JLYYkJuNs#t=90
[Ref.- 136]	"Sharing what's up our sleeve: Android coming to wearables". Official Android Blog. Julio 2014. URL: https://android.googleblog.com/2014/03/sharing-whats-up-our-sleeve-android.html
[Ref.- 137]	"DevBytes: Restricted Profiles in Android 4.3". Google Developers. Julio 2014. URL: https://www.youtube.com/watch?v=pdUcANNm72o
[Ref.- 138]	"CCS Injection Vulnerability". Lepidum. June 2014. URL: http://ccsinjection.lepidum.co.jp URL: http://ccsinjection.lepidum.co.jp/blog/2014-06-05/CCS-Injection-en/index.html
[Ref.- 139]	"Early ChangeCipherSpec Attack". Imperial Violet - Adam Langley. Junio 2014. URL: https://www.imperialviolet.org/2014/06/05/earlyccs.html
[Ref.- 140]	"Security Enhancements in Android x.y". Android Source. URL: https://source.android.com/security/enhancements/index.html URL: https://source.android.com/security/enhancements/enhancements41.html URL: https://source.android.com/security/enhancements/enhancements42.html URL: https://source.android.com/security/enhancements/enhancements43.html URL: https://source.android.com/security/enhancements/enhancements44.html URL: https://source.android.com/security/enhancements/enhancements50.html URL: https://source.android.com/security/enhancements/enhancements60.html
[Ref.- 141]	"Android Hacker's Handbook". Wiley. 2014.
[Ref.- 142]	"Security-Enhanced Linux in Android". Android Source. URL: http://source.android.com/devices/tech/security/se-linux.html
[Ref.- 143]	Google Cloud Messaging (GCM) for Android. Google. URL: http://developer.android.com/google/gcm/index.html

Referencia	Título, autor y ubicación
[Ref.- 144]	"Credential storage enhancements in Android 4.3". Nikolay Elenkov. August 2013. URL: http://nelenkov.blogspot.com.es/2013/08/credential-storage-enhancements-android-43.html
[Ref.- 145]	"Android's 5.x Lock Screen may be bypassed by attackers". Martin Brinkmann. Ghacks.net URL: https://www.ghacks.net/2015/09/16/androids-5-x-lock-screen-may-be-bypassed-by-attackers/
[Ref.- 146]	"Secure USB debugging in Android 4.2.2". Nikolay Elenkov. Febrero 2013. URL: http://nelenkov.blogspot.com.es/2013/02/secure-usb-debugging-in-android-422.html URL: http://android-developers.blogspot.jp/2013/02/security-enhancements-in-jelly-bean.html
[Ref.- 147]	"Utilizar el identificador de llamada de Google". Google. 2014. URL: https://support.google.com/nexus/answer/3459196?hl=es
[Ref.- 148]	"Android 4.4 - KitKat". Google. URL: http://www.android.com/versions/kit-kat-4-4/ URL: https://www.youtube.com/watch?v=sONcojECWxs
[Ref.- 149]	"Administrar la ubicación de tu dispositivo". Google. URL: https://support.google.com/nexus/answer/3467281?hl=es
[Ref.- 150]	Google Play Services. Google. URL: https://developer.android.com/google/play-services/index.html
[Ref.- 151]	"On the WebView addJavascriptInterface Saga". DroidSec. February 2014. URL: http://www.droidsec.org/news/2014/02/26/on-the-webview-addjsif-saga.html
[Ref.- 152]	"Updating Your Security Provider to Protect Against SSL Exploits". Android. URL: http://developer.android.com/training/articles/security-gms-provider.html
[Ref.- 153]	"Issue 52314: unlock password is limited to 16 characters". Android Google Code. February 2013. URL: https://code.google.com/p/android/issues/detail?id=52314
[Ref.- 154]	"Issue 29468: Different passwords for encryption and screen lock". Android Google Code. April 2012. URL: https://code.google.com/p/android/issues/detail?id=29468
[Ref.- 155]	"EncPassChanger". Kibab. December 2012. URL: https://github.com/kibab/encpasschanger URL: https://play.google.com/store/apps/details?id=com.kibab.android.EncPassChanger
[Ref.- 156]	"Cryptfs Password". Nikolay Elenkov. August 2013. URL: https://play.google.com/store/apps/details?id=org.nick.cryptfs.passwdmanager URL: https://github.com/nelenkov/cryptfs-password-manager
[Ref.- 157]	"Changing Android's disk encryption password". Nikolay Elenkov. August 2012. URL: http://nelenkov.blogspot.com/2012/08/changing-androids-disk-encryption.html
[Ref.- 158]	"Android encryption switched to scrypt in 4.4.2". AOSP. URL: https://android.googlesource.com/platform/system/vold/+/_android-4.4_r1/cryptfs.c URL: https://android.googlesource.com/platform/system/vold/+/_log/android-4.4_r1/cryptfs.c (commit c4c70f1) URL: https://android.googlesource.com/platform/system/vold/+/_c4c70f15bb8845b02f9ec1d624794757badd6933
[Ref.- 159]	"Android Security Internals". Nikolay Elenkov. No Starch Press. October 2014. URL: http://www.nostarch.com/androidsecurity
[Ref.- 160]	"Android's Permissions System Is Broken and Google Just Made It Worse". Chris Hoffman. December 2013. URL: http://www.howtogeek.com/177904/androids-permissions-system-is-broken-and-google-just-made-it-worse/
[Ref.- 161]	"Android's App Permissions Were Just Simplified - Now They're Much Less Secure". Chris Hoffman. June 2014. URL: http://www.howtogeek.com/190863/androids-app-permissions-were-just-simplified-now-theyre-much-less-secure/
[Ref.- 162]	"Revisar los permisos de aplicaciones en Android 5.1.1 o anteriores ". Google. 2015. URL: https://support.google.com/googleplay/answer/6014972?hl=es
[Ref.- 163]	"Android 5.x Lockscreen Bypass (CVE-2015-3860)". Austin University of Texas. http://sites.utexas.edu/iso/2015/09/15/android-5-lockscreen-bypass/
[Ref.- 164]	"Añadir aplicaciones, accesos directos y widgets a las pantallas de inicio ". Google. URL: https://support.google.com/nexus/answer/2781850?hl=es

Referencia	Título, autor y ubicación
[Ref.- 165]	"Storage Access Framework". Google. URL: https://developer.android.com/guide/topics/providers/document-provider.html URL: https://www.youtube.com/watch?v=zxHVeXbK1P4 URL: https://www.youtube.com/watch?v=UFj9AEz0DHQ
[Ref.- 166]	"Activar o desactivar SafeSearch". Google. URL: https://support.google.com/websearch/answer/510?hl=es
[Ref.- 167]	"Ubicación en Ajustes de Google". Google. URL: https://support.google.com/accounts/answer/3118687?hl=es
[Ref.- 168]	"ART (Android RunTime)". Google. URL: http://source.android.com/devices/tech/dalvik/
[Ref.- 169]	"Android and Security" (Google Bouncer). Google Mobile Blog. February 2012. URL: http://googlemobile.blogspot.com/2012/02/android-and-security.html
[Ref.- 170]	"Android 4.2 'Verify apps' security feature explained by Google". Android Authority. November 2012. URL: http://www.androidauthority.com/android-4-2-verify-apps-security-feature-explained-by-google-131514/
[Ref.- 171]	"Expanding Google's security services for Android". Android Official Blog. April 2014. URL: http://officialandroid.blogspot.com.es/2014/04/expanding-googles-security-services-for.html
[Ref.- 172]	"Protegerse frente a aplicaciones dañinas". Google. URL: https://support.google.com/accounts/answer/2812853?hl=es URL: https://plus.google.com/+MichaelMorrissey/posts/dutcnbu72pv
[Ref.- 173]	"An Evaluation of the Application ("App") Verification Service in Android 4.2". Xuxian Jiang. December 2012. URL: http://www.cs.ncsu.edu/faculty/jiang/appverify/
[Ref.- 174]	"Dissecting the Android Bouncer". Jon Oberheide. 2012. URL: https://jon.oberheide.org/files/summercon12-bouncer.pdf
[Ref.- 175]	"Actividad web y de aplicaciones". Google. URL: https://support.google.com/accounts/answer/54068?hl=es
[Ref.- 176]	"Eliminar búsquedas y actividad de navegación". Google. URL: https://support.google.com/accounts/answer/465?hl=es
[Ref.- 177]	"Crear una copia de seguridad de tus fotos y vídeos automáticamente". Google. URL: https://support.google.com/plus/answer/1647509
[Ref.- 178]	"Función Reconóceme en fotos y vídeos". Google. URL: https://support.google.com/plus/answer/2370300?hl=es
[Ref.- 179]	"Ver fotos en las que te hayan etiquetado". Google. URL: https://support.google.com/plus/answer/1254833?hl=es
[Ref.- 180]	"Bloquear o desbloquear cuentas de otras personas". Google. URL: https://support.google.com/accounts/answer/6388749?visit_id=1-636343239744150110-383769448&p=block_list&hl=es&rd=1
[Ref.- 181]	"New NFC capabilities through Host Card Emulation". Google. URL: http://developer.android.com/about/versions/kitkat.html#44-hce
[Ref.- 182]	"Cómo compartir contenido con Android Beam". Google. URL: https://support.google.com/nexus/answer/2781895?hl=es
[Ref.- 183]	"Utilizar la función Tocar y pagar con tu dispositivo". Google. URL: https://support.google.com/nexus/answer/3470787?hl=es
[Ref.- 184]	"Security Tips". Google. 2015. URL: http://developer.android.com/training/articles/security-tips.html
[Ref.- 185]	"Utilizar certificados". Google. URL: https://support.google.com/nexus/answer/2844832?hl=es https://developer.android.com/about/versions/nougat/android-7.0-changes#afw
[Ref.- 186]	"ID de publicidad". Google. URL: https://support.google.com/googleplay/android-developer/answer/6048248?hl=es
[Ref.- 187]	"Nexus 5 Stock OTA URLs". XDA Developers. URL: http://forum.xda-developers.com/google-nexus-5/general/ref-nexus-5-stock-ota-urls-t2475327

Referencia	Título, autor y ubicación
[Ref.- 188]	"Using the ICS KeyChain API" (4.0, ICS). Nikolay Elenkov. November 2011. URL: http://nelenkov.blogspot.com.es/2011/11/using-ics-keychain-api.html "ICS Credential Storage Implementation". Part 1 & 2. November/December 2011. URL: http://nelenkov.blogspot.com.es/2011/11/ics-credential-storage-implementation.html URL: http://nelenkov.blogspot.com.es/2011/12/ics-credential-storage-implementation.html
[Ref.- 189]	"ICS Trust Store Implementation" (4.0, ICS). Nikolay Elenkov. December 2011. URL: http://nelenkov.blogspot.com.es/2011/12/ics-trust-store-implementation.html
[Ref.- 190]	"Certificate blacklisting in Jelly Bean" (4.1, JB). Nikolay Elenkov. July 2012. URL: http://nelenkov.blogspot.com.es/2012/07/certificate-blacklisting-in-jelly-bean.html
[Ref.- 191]	"Jelly Bean hardware-backed credential storage" (4.1, JB). Nikolay Elenkov. July 2012. URL: http://nelenkov.blogspot.jp/2012/07/jelly-bean-hardware-backed-credential.html
[Ref.- 192]	"Questioning the chain of trust: investigations into the root certificates on mobile devices". Andrew Blach. Bluebox. October 2014. URL: https://bluebox.com/blog/technical/questioning-the-chain-of-trust-investigations-into-the-root-certificates-on-mobile-devices/
[Ref.- 193]	"Iniciar sesión con tu teléfono en lugar de usar una contraseña". Google. URL: https://support.google.com/accounts/answer/6361026?hl=es
[Ref.- 194]	"Why Do Wi-Fi Clients Disclose their PNL for Free Still Today?". Raúl Siles. DinoSec. February 2015. URL: http://blog.dinosec.com/2015/02/why-do-wi-fi-clients-disclose-their-pnl.html
[Ref.- 195]	"A Systematic Security Evaluation of Android's Multi-User Framework". 2014. URL: http://www.cis.syr.edu/~wedu/Research/paper/multi_user_most2014.pdf
[Ref.- 196]	"Android Internals: A Confectioner's Cookbook". Jonathan Levin. 2015. URL: http://newandroidbook.com URL: http://newandroidbook.com/files/Andevcon-ART.pdf ("ART (& OAT) Internals")
[Ref.- 197]	"Android for Work". Android. URL: https://www.android.com/work/ URL: https://enterprise.google.com/android
[Ref.- 198]	"Samsung Android Security Updates". Samsung. October 2015. URL: http://security.samsungmobile.com/smrupdate.html
[Ref.- 199]	"Full OTA Images for Nexus and Pixel Devices". Google. URL: https://developers.google.com/android/ota
[Ref.- 200]	"Application Signing". Android Source. URL: https://source.android.com/security/apksigning/
[Ref.- 201]	"Encontrar un dispositivo Android que has perdido". Google. https://support.google.com/accounts/answer/3265955?hl=es
[Ref.- 202]	"Android History". Android. URL: https://www.android.com/history/
[Ref.- 203]	"Android Security Center". Android. URL: https://www.android.com/intl/es_es/security-center/monthly-security-updates/
[Ref.- 204]	"Security Updates and Resources". Android. URL: https://source.android.com/security/overview/updates-resources
[Ref.- 205]	"All together now. Introducing G Suite.". Google Cloud Official blog. September 2016. URL: https://cloud.googleblog.com/2016/09/all-together-now-introducing-G-Suite.html
[Ref.- 206]	"Dissecting Lollipop's Smart Lock". Nelenkov. December 2014. URL: https://nelenkov.blogspot.com.es/2014/12/dissecting-lollipops-smart-lock.html
[Ref.- 207]	Android Security Bulletin. Google. URL: https://source.android.com/security/bulletin/2016-12-01
[Ref.- 208]	"How many combinations does Android 9 point unlock have?". Quora. February 2011. URL: https://www.quora.com/Android-operating-system-How-many-combinations-does-Android-9-point-unlock-have/answer/Yoyo-Zhou URL: http://beust.com/weblog2/archives/000497.html
[Ref.- 209]	"Password storage in Android M". Nikolay Elenkov. June 2015. URL: https://nelenkov.blogspot.com.es/2015/06/password-storage-in-android-m.html
[Ref.- 210]	"Añadir aplicaciones, accesos directos y widgets a las pantallas de inicio". Google. URL: https://support.google.com/nexus/answer/2781850?hl=es

Referencia	Título, autor y ubicación
[Ref.- 211]	"Administrar invitados y usuarios". Google. URL: https://support.google.com/nexus/answer/2865944?hl=es
[Ref.- 212]	"Utilizar perfiles restringidos en tablets". Ayuda de Nexus. URL: https://support.google.com/nexus/answer/3175031
[Ref.- 213]	"Configurar la gestión de dispositivos móviles". Ayuda de G-Suite. Google. URL: https://support.google.com/a/answer/6328699?hl=es
[Ref.- 214]	"The Chromium Projects". URL: https://www.chromium.org/
[Ref.- 215]	"Informe de transparencia". Google. URL: https://www.google.com/transparencyreport/safebrowsing/?hl=es-419
[Ref.- 216]	"Administrar advertencias sobre sitios web no seguros". Google. URL: https://support.google.com/chrome/answer/99020?co=GENIE.Platform%3DAndroid
[Ref.- 217]	"Modificar la configuración de contenido de los sitios web". Google. URL: https://support.google.com/chrome/answer/114662?co=GENIE.Platform%3DAndroid&hl=es
[Ref.- 218]	"Conectar un sitio web a un dispositivo Bluetooth o USB". Google. URL: https://support.google.com/chrome/answer/6362090
[Ref.- 219]	"Navegar en privado con el modo incógnito". Google. URL: https://support.google.com/chrome/answer/95464?co=GENIE.Platform%3DAndroid&hl=es
[Ref.- 220]	"QUIC, a multiplexed stream transport over UDP". "The Chromium project". URL: https://www.chromium.org/quic
[Ref.- 221]	"PdfRenderer". Android Developers. URL: https://developer.android.com/reference/android/graphics/pdf/PdfRenderer.html
[Ref.- 222]	"Actualizar las aplicaciones Android". Ayuda de Google Play. URL: https://support.google.com/googleplay/answer/113412?hl=es
[Ref.- 223]	"Advisory: Crashing Android devices with large Assisted-GPS Data Files [CVE-2016-5348]". Nightwatch Cybersecurity. October 2016. URL: https://www.nightwatchcybersecurity.com/2016/10/04/advisory-cve-2016-5348-2/
[Ref.- 224]	"USB Host and Accessory". Android developers. URL: https://developer.android.com/guide/topics/connectivity/usb/index.html
[Ref.- 225]	"Android HAX". Jon Oberheide. URL: https://jon.oberheide.org/files/summercon10-androidhax-jonoberheide.pdf
[Ref.- 226]	"Android Device Manager has a new name: Find My Device". Engadget. URL: https://www.engadget.com/2017/05/18/google-lost-android-find-my-device/
[Ref.- 227]	"Descargar zonas y navegar sin conexión". Google. URL: https://support.google.com/maps/answer/6291838?co=GENIE.Platform%3DAndroid&hl=es
[Ref.- 228]	"Ver o editar tu cronología". Google. URL: https://support.google.com/maps/answer/6258979?hl=es&ref_topic=3092425&co=GENIE.Platform%3DAndroid&oco=1
[Ref.- 229]	"Ayuda de Google Maps". Google. URL: https://support.google.com/maps#topic=3092425
[Ref.- 230]	"Share your trips and real-time location from Google Maps". Google blog. URL: https://www.blog.google/products/maps/share-your-trips-and-real-time-location-google-maps/
[Ref.- 231]	"Compartir tu ubicación en tiempo real con otros usuarios". Ayuda de Google Maps. URL: https://support.google.com/maps/answer/7326816?hl=es&ref_topic=3092425&co=GENIE.Platform%3DAndroid&oco=1
[Ref.- 232]	"Iniciar sesión en Chrome". Google. URL: https://support.google.com/chrome/answer/185277?co=GENIE.Platform%3DDesktop&hl=es
[Ref.- 233]	"Compartir Chrome con otros usuarios o añadir un perfil". Google. URL: https://support.google.com/chrome/answer/2364824?co=GENIE.Platform%3DAndroid&oco=1
[Ref.- 234]	"Protecting hundreds of millions more mobile users". Google Security Blog. URL: https://security.googleblog.com/2015/12/protecting-hundreds-of-millions-more.html
[Ref.- 235]	"Gestionar la opción Información de los dispositivos". Google. URL: https://support.google.com/accounts/answer/6135999?p=account_device_info&hl=es&authuser=0&visit_id=1-636342504159902489-3147238312&rd=1

Referencia	Título, autor y ubicación
[Ref.- 236]	"Cómo fijar y dejar de fijar pantallas". Google. URL: https://support.google.com/nexus/answer/6118421?hl=es-419
[Ref.- 237]	"Hardware-accelerated disk encryption in Android 5.1". Nikolay Elenkov. May 2015. URL: https://nelenkov.blogspot.com.es/2015/05/hardware-accelerated-disk-encryption-in.html
[Ref.- 238]	"Sincronizar contraseñas en tus dispositivos". Google. URL: https://support.google.com/accounts/answer/6197437?co=GENIE.Platform%3DAndroid
[Ref.- 239]	"Android Instant Apps". Google Developer. URL: https://developer.android.com/topic/instant-apps/index.html
[Ref.- 240]	"Presentación de las Instant Apps en Google I/O 2016". Google. URL: https://youtu.be/cosqlfqrpFA
[Ref.- 241]	"Android Instant Apps: The security jury is out". NowSecure. URL: https://www.nowsecure.com/blog/2016/05/19/android-instant-apps-the-security-jury-is-out/
[Ref.- 242]	"Will Google's Instant Apps Undermine Enterprise Security?". Appthority. URL: https://www.appthority.com/mobile-threat-center/blog/will-googles-instant-apps-undermine-enterprise-security/
[Ref.- 243]	"Sincronizar contraseñas en tus dispositivos". Google. URL: https://support.google.com/accounts/answer/6197437?co=GENIE.Platform%3DAndroid&oco=1
[Ref.- 244]	"Introducing the Google Identity Platform". Google developers blog. URL: https://developers.googleblog.com/2015/05/introducing-google-identity-platform.html
[Ref.- 245]	"Smart Lock for Passwords on Android". Google developers. URL: https://developers.google.com/identity/smartlock-passwords/android/
[Ref.- 246]	"Android One". Google. URL: https://www.android.com/one/
[Ref.- 247]	"Taking the next step with Android One". Google. URL: https://www.blog.google/products/android/taking-next-step-android-one/
[Ref.- 248]	"Cómo administrar tus claves de firma de apps". Google. URL: https://support.google.com/googleplay/android-developer/answer/7384423?hl=es-419
[Ref.- 249]	"Security-Enhanced Linux in Android". Google. URL: https://source.android.com/security/selinux/
[Ref.- 250]	"KeyChain". Android Developers. URL: https://developer.android.com/reference/android/security/KeyChain.html
[Ref.- 251]	"Google collects Android users' locations even when location services are disabled". Quartz. URL: https://qz.com/1131515/google-collects-android-users-locations-even-when-location-services-are-disabled/
[Ref.- 252]	"Google Privacy Policy". Google. URL: https://www.google.com/policies/privacy/
[Ref.- 253]	"iBeacon". Apple. URL: https://developer.apple.com/ibeacon/
[Ref.- 254]	"Platform Overview". Google Beacon Platform. URL: https://developers.google.com/beacons/
[Ref.- 255]	"G Pay". Google. URL: https://www.blog.google/topics/shopping-payments/announcing-google-pay/
[Ref.- 256]	"Legacy Core Specifications". Bluetooth SIG. URL: https://www.bluetooth.com/specifications/bluetooth-core-specification/legacy-specifications
[Ref.- 257]	"Wireshark". Wireshark.org. URL: https://www.wireshark.org/
[Ref.- 258]	"Firebase". Google. URL: https://firebase.google.com/docs
[Ref.- 259]	"OAuth 2.0". OAuth.net. URL: http://oauth.net/2/
[Ref.- 260]	"Instalar Google Authenticator". Google. URL: https://support.google.com/accounts/answer/1066447
[Ref.- 261]	"Utilizar la llave de seguridad para la verificación en dos pasos". Google. URL: https://support.google.com/accounts/answer/6103523?hl=es

Referencia	Título, autor y ubicación
[Ref.- 262]	¿Qué es el centro de pagos de Google?. Google. URL: https://support.google.com/payments/answer/7500996
[Ref.- 263]	"Add Google Sign-In to Your Android App". Google. URL: https://developers.google.com/identity/sign-in/android/
[Ref.- 264]	"Files that are backed up". Google. URL: https://developer.android.com/guide/topics/data/autobackup.html#Files
[Ref.- 265]	"Back Up Key-Value Pairs with Android Backup Service". Google. URL: https://developer.android.com/guide/topics/data/keyvaluebackup.html
[Ref.- 266]	"Inhabilitar el inicio de sesión automático". Google. URL: https://support.google.com/accounts/answer/39273?hl=es
[Ref.- 267]	"DigiNotar Certificate Authority Breach Crashes e-Government in the Netherlands". IEEE Spectrum. URL: https://spectrum.ieee.org/riskfactor/telecom/security/diginotar-certificate-authority-breach-crashes-egovernment-in-the-netherlands
[Ref.- 268]	"Device Administration". Android Developers. URL: https://developer.android.com/guide/topics/admin/device-admin.html
[Ref.- 269]	"Google Digital Asset Links". Google Developers. URL: https://developers.google.com/digital-asset-links/v1/getting-started
[Ref.- 270]	"The Android Trojan Svpeng now capable of Mobile Phishing". SecureList. URL: https://securelist.com/the-android-trojan-svpeng-now-capable-of-mobile-phishing/57301/ "The evolution of Acecard". SecureList. URL: https://securelist.com/the-evolution-of-acecard/73777/
[Ref.- 271]	"The best antivirus software for Android". The Independent IT-Security Institute. URL: https://www.av-test.org/en/antivirus/mobile-devices/android/september-2017/?_ga=2.224379036.604887240.1508931778-1860796072.1508931778
[Ref.- 272]	"android.app.admin overview". Android developers. URL: https://developer.android.com/reference/android/app/admin/package-summary.html
[Ref.- 273]	"Chromium OS". Google. URL: https://www.chromium.org/chromium-os
[Ref.- 274]	"Utilizar Nearby para interactuar con lo que te rodea". Ayuda de cuenta de Google. URL: https://support.google.com/accounts/answer/6260286?hl=es
[Ref.- 275]	"Gatekeeper Password Storage: How Android Secures Devices". Magnet Forensics. URL: https://www.magnetforensics.com/blog/gatekeeper-password-storage-android-secures-devices/
[Ref.- 276]	"Cracking Android Pattern Lock in Five Attempts". School of Information Science and Technology, Northwest University, China. School of Computing and Communications, Lancaster University, UK. URL: http://www.lancaster.ac.uk/staff/wangz3/publications/ndss_17.pdf
[Ref.- 277]	"Adding an easy share action". Android Developers. URL: https://developer.android.com/training/sharing/shareaction
[Ref.- 278]	"NFC Simple NDEF Exchange Protocol Specification". NFC Forum. URL: https://nfc-forum.org/product/nfc-simple-ndef-exchange-protocol-specification/
[Ref.- 279]	"Empezar a utilizar TalkBack en Android". Google. URL: https://support.google.com/accessibility/android/answer/6283677?hl=es
[Ref.- 280]	"FPR vulnerada". Varios. NIST: URL: https://nvd.nist.gov/nvd.cfm?cvename=CVE-2017-8152 Android Police: URL: https://www.androidpolice.com/2015/11/02/rootjunky-shows-off-just-how-easy-it-is-to-bypass-the-factory-reset-protection-on-samsung-phones-due-to-a-flaw-in-their-software/ URL: https://www.androidpolice.com/2016/01/12/lgs-factory-reset-protection-can-be-bypassed-in-mere-minutes/
[Ref.- 281]	"Nearby". Android Developers. URL: https://developers.google.com/nearby/
[Ref.- 282]	"Nearby Connections API". URL: https://developers.google.com/nearby/connections/overview#introduction

La siguiente tabla muestra las fuentes de información a las que se hace referencia a lo largo de la presente guía, en concreto las referencias asociadas directamente al CCN-CERT:

Referencia	Título, autor y ubicación
[Ref.- 400]	"Guía CCN-STIC-450: Seguridad de dispositivos móviles". CCN-CERT. Marzo 2013. URL: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/6-ccn-stic-450-seguridad-en-dispositivos-moviles/file.html
[Ref.- 401]	"Guía CCN-STIC-453(A) - Seguridad de dispositivos móviles: Android 2.x". CCN-CERT. Mayo 2013. URL: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/11-ccn-stic-453-seguridad-en-android/file.html
[Ref.- 402]	"Guía CCN-STIC-453B - Seguridad de dispositivos móviles: Android 4.x". CCN-CERT. Abril 2015. URL: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/827-ccn-stic-453b-seguridad-en-android-4-x/file.html
[Ref.- 403]	"CCN-STIC-457: Gestión de dispositivos móviles: MDM (Mobile Device Management)". CCN-CERT. Noviembre 2013. URL: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/14-ccn-stic-457-herramienta-de-gestion-de-dispositivos-moviles-mdm/file.html
[Ref.- 404]	"Buenas Prácticas. CCN-CERT BP-03/16. Dispositivos móviles". CCN-CERT. Octubre 2016. URL: https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/1757-ccn-cert-bp-03-16-dispositivos-moviles/file.html
[Ref.- 405]	"Guía CCN-STIC-453C - Seguridad de dispositivos móviles: Android 5.x". CCN-CERT. Enero 2018. URL: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/2733-ccn-stic-453c-seguridad-en-android-5-x.html
[Ref.- 406]	"Guía CCN-STIC-453D - Seguridad de dispositivos móviles: Android 6.x". CCN-CERT. Mayo 2018. URL: https://www.ccn-cert.cni.es/series-ccn-stic/guias-de-acceso-publico-ccn-stic/2901-ccn-stic-453d-seguridad-de-dispositivos-moviles-android-6-x.html
[Ref.- 407]	"Guía CCN-STIC-456 - Guía de cuenta de usuario, servicios y aplicaciones de Google para dispositivos móviles Android". CCN-CERT. SEP 2018.

La siguiente tabla muestra las fuentes de información a las que se hace referencia a lo largo de la presente guía relacionadas específicamente con la versión 5.x (Lollipop) de Android:

Referencia	Título, autor y ubicación
[Ref.- 500]	"Android 5.0 Lollipop". Android. URL: https://www.android.com/versions/lollipop-5-0/
[Ref.- 501]	"Android Lollipop". Android Developers. URL: https://developer.android.com/about/versions/lollipop.html
[Ref.- 502]	"Android 5.0 APIs". Android Developers. URL: https://developer.android.com/about/versions/android-5.0.html
[Ref.- 503]	"Android 5.0 Behavior Changes". Android Developers. URL: https://developer.android.com/about/versions/android-5.0-changes.html
[Ref.- 504]	"Android 5.1 APIs". Android Developers. URL: https://developer.android.com/about/versions/android-5.1.html
[Ref.- 505]	"Android in the Workplace and in Education". Android 5.0 Developers. URL: https://developer.android.com/about/versions/android-5.0.html#Enterprise
[Ref.- 506]	"Android Platform Tools r20". URL: http://dl-ssl.google.com/android/repository/platform-tools_r20-macosx.zip URL: http://dl-ssl.google.com/android/repository/platform-tools_r20-windows.zip URL: http://dl-ssl.google.com/android/repository/platform-tools_r20-linux.zip
[Ref.- 507]	"Notifications". Android Developers - Google. URL: https://developer.android.com/guide/topics/ui/notifiers/notifications.html

Referencia	Título, autor y ubicación
[Ref.- 508]	"Android developers". Google. URL: https://developer.android.com/about/versions/android-5.0-changes.html#BehaviorNotifications
[Ref.- 509]	"Android Official Blog". Google. URL: https://android.googleblog.com/2015/03/android-5.1-unwrapping-new-lollipop.html
[Ref.- 510]	"Memory Leak on Lollipop crashing Apps" - Google issue tracker. https://issuetracker.google.com/issues/37010796
[Ref.- 511]	"UICC Carrier Privileges". Android Source. URL: https://source.android.com/devices/tech/config/uicc
[Ref.- 512]	"Verified Boot". Android. URL: https://source.android.com/security/verifiedboot/ URL: https://source.android.com/security/verifiedboot/verified-boot.html
[Ref.- 513]	"Reiniciar en modo seguro para detectar aplicaciones con problemas". Soporte de Google. URL: https://support.google.com/nexus/answer/2852139?hl=es
[Ref.- 514]	"Cambio de nombre de Android for Work a Android y de Google Play for Work a Google Play". Google. URL: https://support.google.com/work/android/answer/7218437 URL: https://blog.google/topics/connected-workspaces/update-android-and-google-plays-progress-enterprise/
[Ref.- 515]	"Screen Pinning (Fijación de pantalla)". Android Developers. URL: https://developer.android.com/about/versions/android-5.0.html#ScreenPinning
[Ref.- 516]	"¿Qué es un perfil de trabajo?". Google. URL: https://support.google.com/work/android/answer/6191949
[Ref.- 517]	"Tricking Android Smart Lock with Bluetooth". Martin Herfurt. May 2015. URL: https://mherfurt.wordpress.com/2015/05/08/tricking-android-smart-lock-with-bluetooth/ URL: https://youtu.be/Xb0YG6EpY48
[Ref.- 518]	"Set up your device for automatic unlock". Google. Nexus Help. URL: https://support.google.com/nexus/answer/6093922?hl=en
[Ref.- 519]	"Consultar el estado de certificación del dispositivo". Ayuda de Google Play. URL: https://support.google.com/googleplay/answer/7165974?hl=es
[Ref.- 520]	"Device Policy Manager - Password quality". Android Developers. URL: https://developer.android.com/reference/android/app/admin/DevicePolicyManager.html#setPasswordQuality(android.content.ComponentName,%20int)
[Ref.- 521]	"Evitar que otros usuarios utilicen tu dispositivo sin permiso". Google. URL: https://support.google.com/nexus/answer/6172890?hl=es

La siguiente tabla muestra las fuentes de información a las que se hace referencia a lo largo de la presente guía relacionadas específicamente con la versión 6.x (Marshmallow) de Android:

Referencia	Título, autor y ubicación
[Ref.- 600]	"Android 6.0 Marshmallow". Android. URL: https://www.android.com/versions/marshmallow-6-0/
[Ref.- 601]	"Android 6.0 Marshmallow". Android Developers. URL: https://developer.android.com/about/versions/marshmallow/index.html
[Ref.- 602]	"Android 6.0 Changes". Android Developers. URL: https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html
[Ref.- 603]	"Android 6.0 APIs". Android Developers. URL: https://developer.android.com/about/versions/marshmallow/android-6.0.html
[Ref.- 604]	"Handling App Links". Android Developers. URL: https://developer.android.com/training/app-links/index.html

Referencia	Título, autor y ubicación
[Ref.- 605]	"BoringSSL". Google. URL: https://boringssl.googlesource.com/boringssl/ URL: https://github.com/google/boringssl URL: https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html?#boringSSL
[Ref.- 606]	"Android (Marshmallow) Compatibility Definition Document (CDD)" (7.3.10. Fingerprint Sensor). Google. URL: https://source.android.com/compatibility/6.0/android-6.0-cdd
[Ref.- 607]	"Fingerprint and payments APIs (100 Days of Google Dev)". Google Developers. May 2015. URL: https://www.youtube.com/watch?v=VOn7VrTRIA4
[Ref.- 608]	"Fingerprint security on Nexus devices". Google Support. URL: https://support.google.com/nexus/answer/6300638
[Ref.- 609]	"Desbloqueo mediante huella dactilar digital". Google Support. URL: https://support.google.com/nexus/answer/6285273
[Ref.- 610]	"Fingerprint Unlock Security: iOS vs. Google Android (Part I & II)". Elcomsoft. June 2016. URL: http://blog.elcomsoft.com/2016/06/ios-fingerprint-unlock-security/ URL: http://blog.elcomsoft.com/2016/06/fingerprint-unlock-security-ios-vs-google-android-part-ii/
[Ref.- 611]	"Fingerprints On Mobile Devices: Abusing and Leaking". FireEye Labs. BH USA 2015. URL: https://www.blackhat.com/docs/us-15/materials/us-15-Zhang-Fingerprints-On-Mobile-Devices-Abusing-And-Leaking-wp.pdf
[Ref.- 612]	"App Install Location". Android Developers. URL: https://developer.android.com/guide/topics/data/install-location.html
[Ref.- 613]	"Apps on SD Card: The Details". Android Developers Blog. July 2010. URL: http://android-developers.blogspot.com/2010/07/apps-on-sd-card-details.html
[Ref.- 614]	"Decrypting Android M adopted storage". Nikolay Elenkov. Android Explorations. June 2015. URL: https://nelenkov.blogspot.com.es/2015/06/decrypting-android-m-adopted-storage.html
[Ref.- 615]	"Fingerprint HAL". Android Source. URL: https://source.android.com/security/authentication/fingerprint-hal.html
[Ref.- 616]	"Optimizing for Doze and App Standby". Android Developers. URL: https://developer.android.com/training/monitoring-device-state/doze-standby.html
[Ref.- 617]	"Requesting Permissions at Run Time". Android Developers. URL: https://developer.android.com/training/permissions/requesting.html
[Ref.- 618]	"Cómo trabajar con permisos del sistema". Android Developers. URL: https://developer.android.com/training/permissions/index.html URL: https://developer.android.com/training/permissions/best-practices.html#testing
[Ref.- 619]	"Warning about operating system safety". Nexus. URL: http://g.co/ABH URL: https://support.google.com/nexus/answer/6185381?p=verified_boot
[Ref.- 620]	"Password storage in Android M". Nikolay Elenkov. Android Explorations. June 2015. URL: https://nelenkov.blogspot.com.es/2015/06/password-storage-in-android-m.html
[Ref.- 621]	"Hardware-backed Keystore". Google. URL: https://source.android.com/security/keystore/
[Ref.- 622]	"Trusty TEE". Android Security. URL: https://source.android.com/security/trusty/
[Ref.- 623]	"Controlar los permisos de una aplicación en Android 6.0 y versiones superiores". Ayuda de Google Play. URL: https://support.google.com/googleplay/answer/6270602
[Ref.- 624]	"Cómo activar o desactivar tu feed". Google. URL: https://support.google.com/websearch/answer/2824784?hl=es-419&co=GENIE.Platform=Android
[Ref.- 625]	"Usar Mi tablón en la aplicación de Google". Google. URL: https://support.google.com/websearch/answer/2819496?hl=es
[Ref.- 626]	"Personalizar el Tablón de la aplicación de Google". Google. URL: https://support.google.com/websearch/answer/6237870?&co=GENIE.Platform%3DAndroid
[Ref.- 627]	"Requesting Permissions". Google developers. URL: https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous

Referencia	Título, autor y ubicación
[Ref.- 628]	"Lock screen notifications". Android developers. URL: https://developer.android.com/guide/topics/ui/notifiers/notifications.html#lockscreenNotification
[Ref.- 629]	"Keystore redesign in Android M". Nelenkov. June 2015. URL: https://nelenkov.blogspot.com.es/2015/06/keystore-redesign-in-android-m.html
[Ref.- 630]	"Android Keystore System". Android developers. URL: https://developer.android.com/training/articles/keystore.html
[Ref.- 631]	GateKeeper. "Android Source". URL: https://source.android.com/security/authentication/gatekeeper
[Ref.- 632]	"Bluetooth Services". Android Source. URL: https://source.android.com/devices/bluetooth/services
[Ref.- 633]	"Find people, things, & places in your photos". Google. URL: https://support.google.com/photos/answer/6128838
[Ref.- 634]	"Back Up User Data with Auto Backup". Android Developers. URL: https://developer.android.com/guide/topics/data/autobackup.html
[Ref.- 635]	"Auto Backup for Apps made simple". Android developers blog. URL: https://android-developers.googleblog.com/2015/07/auto-backup-for-apps-made-simple.html

La siguiente tabla muestra las fuentes de información a las que se hace referencia a lo largo de la presente guía relacionadas específicamente con la versión 7.x (Nougat) de Android:

Referencia	Título, autor y ubicación
[Ref.- 700]	"Android 7.0 Nougat". Android. URL: https://www.android.com/versions/nougat-7-0/
[Ref.- 701]	"Android 7.0 Nougat". Android Developers. URL: https://developer.android.com/about/versions/nougat/index.html
[Ref.- 702]	"Android 7.0 for Developers". Android Developers. URL: https://developer.android.com/about/versions/nougat/android-7.0.html
[Ref.- 703]	"Android 7.0 Behavior Changes". Android Developers. URL: https://developer.android.com/about/versions/nougat/android-7.0-changes.html
[Ref.- 704]	"Pixel y Pixel XL". Made by Google. October 2016. URL: https://madeby.google.com/phone/
[Ref.- 705]	"Six years of Nexus: A Google Phone History". Android Police. September 2016. URL: https://www.youtube.com/watch?v=7_tAJlm6xA
[Ref.- 706]	"Keeping Android safe: Security enhancements in Nougat". Android Developers Blog. Sep 2016. URL: http://android-developers.blogspot.com.es/2016/09/security-enhancements-in-nougat.html
[Ref.- 707]	"Hardening the media stack". Android Developers Blog. May 2016. URL: http://android-developers.blogspot.com/2016/05/hardening-media-stack.html
[Ref.- 708]	Seccomp. WikiPedia. URL: https://en.wikipedia.org/wiki/Seccomp
[Ref.- 709]	"Security "Crypto" provider deprecated in Android N". Android Developers Blog. June 2016. URL: https://android-developers.blogspot.com.es/2016/06/security-crypto-provider-deprecated-in.html
[Ref.- 710]	"Improving Stability with Private C/C++ Symbol Restrictions in Android N". Android Developers Blog. June 2016. URL: https://android-developers.blogspot.com.es/2016/06/improving-stability-with-private-cc.html
[Ref.- 711]	"Changes to Trusted Certificate Authorities in Android Nougat". Android Developers Blog. July 2016. URL: https://android-developers.blogspot.com.es/2016/07/changes-to-trusted-certificate.html

Referencia	Título, autor y ubicación
[Ref.- 712]	"APK Signature Scheme v2". Android. URL: https://source.android.com/security/apksigning/v2.html URL: https://android-developers.googleblog.com/2016/11/understanding-apk-packaging-in-android-studio-2-2.html
[Ref.- 713]	Android Security Center. Google. November 2016. URL: https://www.android.com/security-center/
[Ref.- 714]	"The Google Android Security Team's Classifications for Potentially Harmful Applications". Android security Center. April 2016. URL: https://static.googleusercontent.com/media/source.android.com/en//security/reports/Google_Android_Security_PHA_classifications.pdf
[Ref.- 715]	"File System/Autoupdate". The Chromium projects. URL: https://www.chromium.org/chromium-os/chromiumos-design-docs/filesystem-autoupdate
[Ref.- 716]	"Stagefright (bug)". Wikipedia. URL: https://en.wikipedia.org/wiki/Stagefright_(bug)
[Ref.- 717]	"How we keep harmful apps out of Google Play and keep your Android device safe". Android security Center. February 2016. URL: https://static.googleusercontent.com/media/source.android.com/en//security/reports/Android_WhitePaper_Final_02092016.pdf
[Ref.- 718]	"Android 7.1, (N) Compatibility Definition Document - 7.3.10. Fingerprint Sensor". Google. https://source.android.com/compatibility/7.1/android-7.1-cdd
[Ref.- 719]	"Bloquear o desbloquear un número de teléfono". Google. URL: https://support.google.com/nexus/answer/6325463
[Ref.- 720]	"Utilizar el ID de llamada y la protección contra spam". Google. URL: https://support.google.com/nexus/answer/3459196?hl=es
[Ref.- 721]	"Factory Images and Full OTA Images for Android 7.0". XDA Developers. https://www.xda-developers.com/factory-images-and-full-ota-images-for-android-7-0-nougat-for-nexus-going-live-updated-with-nexus-6p-links/ "7.1.2 OTA and device Images now available". XDA Developers. https://forum.xda-developers.com/pixel-xl/help/7-1-2-ota-device-images-available-t3583895
[Ref.- 722]	"Protecting Android with more Linux kernel defenses". Android Developers Blog. July 2016. URL: http://android-developers.blogspot.com/2016/07/protecting-android-with-more-linux.html
[Ref.- 723]	"Strictly Enforced Verified Boot with Error Correction". Android Developers Blog. July 2016. URL: http://android-developers.blogspot.com/2016/07/strictly-enforced-verified-boot-with.html
[Ref.- 724]	"Strict Mode". Android Developers. URL: https://developer.android.com/reference/android/os/StrictMode.html
[Ref.- 725]	"Partition selection (slots)". Android Source. URL: https://source.android.com/devices/tech/ota/ab/#slots
[Ref.- 726]	"A/B (Seamless) System Updates". Android Source. URL: https://source.android.com/devices/tech/ota/ab/
[Ref.- 727]	"Implementing dm-verity". Android Source. URL: https://source.android.com/security/verifiedboot/dm-verity
[Ref.- 728]	"Verified Boot". The Chromium projects. URL: http://www.chromium.org/chromium-os/chromiumos-design-docs/verified-boot
[Ref.- 729]	"Build a device policy controller". Android developers. URL: https://developer.android.com/work/dpc/build-dpc.html
[Ref.- 730]	"Android 7.0 Behavior Changes - Android for work". Android developers. URL: https://developer.android.com/about/versions/nougat/android-7.0-changes.html#afw
[Ref.- 731]	"Work profile". Android EMM Developers. URL: https://developers.google.com/android/work/requirements/work-profile
[Ref.- 732]	"Work profile security challenge". Android developers. URL: https://developer.android.com/work/dpc/security.html#work_profile_security_challenge

Referencia	Título, autor y ubicación
[Ref.- 733]	"Here's a list of Android devices that support Seamless Updates". XDA-Developers. URL: https://www.xda-developers.com/list-android-devices-seamless-updates/
[Ref.- 734]	"File-Based Encryption". Android Source. URL: https://source.android.com/security/encryption/file-based
[Ref.- 735]	"Supporting Direct Boot". Android developers. https://developer.android.com/training/articles/direct-boot.html
[Ref.- 736]	"Security-Enhanced Linux in Android". Android Source. URL: https://source.android.com/security/selinux/device-policy
[Ref.- 737]	"SELinux concepts". Android Source. URL: https://source.android.com/security/selinux/concepts
[Ref.- 738]	"Nexus Security Bulletin - March 2016". Android Security. URL: https://source.android.com/security/bulletin/2016-03-01#elevation_of_privilege_vulnerability_in_mediatek_wi-fi_kernel_driver
[Ref.- 739]	"ioctl command whitelisting in SELinux". Jeff Vander Stoep. URL: http://kernsec.org/files/lss2015/vanderstoep.pdf
[Ref.- 740]	"Protecting Android with more Linux kernel defenses". Android Developers Blog. URL: https://android-developers.googleblog.com/2016/07/protecting-android-with-more-linux.html
[Ref.- 741]	"Protecting Android with more Linux kernel defenses". Android Developers Blog. URL: https://android-developers.googleblog.com/2016/07/protecting-android-with-more-linux.html
[Ref.- 742]	"How does Android save your fingerprints?". Android Central. URL: https://www.androidcentral.com/how-does-android-save-your-fingerprints
[Ref.- 743]	"Trusty API Reference". Android Source. URL: https://source.android.com/security/trusty/trusty-ref
[Ref.- 744]	"[PSA - Update] Google breaks their silence and gives an Official reason for the Removal of NFC Smart Unlock on Android". Reddit. URL: https://www.reddit.com/r/Android/comments/73ejrk/psa_update_google_breaks_their_silence_and_gives/
[Ref.- 745]	"Key Generator". Android developers. URL: https://developer.android.com/training/articles/keystore?hl=es-419#SupportedKeyGenerators
[Ref.- 746]	"Add protection against LE scanning abuse". Android Open Source Project. May 2016. URL: https://android-review.googlesource.com/c/platform/packages/apps/Bluetooth/+215844
[Ref.- 747]	"Network security configuration". Android Developers. URL: https://developer.android.com/training/articles/security-config
[Ref.- 748]	"Secure boot and image authentication in mobile tech". Qualcomm. January 2017. URL: https://www.qualcomm.com/news/onq/2017/01/17/secure-boot-and-image-authentication-mobile-tech