centro criptológico nacional

Guía de Seguridad de las TIC
CCN-STIC 481

Buenas Prácticas para Codificación Segura



Noviembre 2026

MINISTERIO
DE DEFENSA

cpage.mpr.gob.es

Catálogo de Publicaciones de la Administración General del Estado
cpage.mpr.gob.es

Edita:

MINISTERIO DE DEFENSA
SUBSECRETARÍA DE DEFENSA
SECRETARÍA GENERAL TÉCNICA

CCN
centro criptológico nacional

Pº de la Castellana 109, 28046 Madrid
© Centro Criptológico Nacional, 2026
NIPO (edición en línea)

CCN
c=ES, l=MADRID, o=CCN,
ou=SELLO ELECTRONICO,
2.5.4.97=VATES-S2800155J,
serialNumber=S2800155J,
cn=CCN
2026.11.23 11:35:51 +01:00

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.

Esta publicación se distribuye bajo licencia CC BY-NC-ND 4.0 que permite compartir el material en cualquier medio o formato con la condición de reconocer adecuadamente la procedencia «Edita: Ministerio de Defensa. Secretaría General Técnica.» No se puede modificar. No se puede utilizar con fines comerciales.

ÍNDICE

II

1. INTRODUCCIÓN 1

1.1 OBJETIVO 1

1.2 ALCANCE 1

2. PRINCIPIOS GENERALES DE CODIFICACIÓN SEGURA 3

2.1 SEGURIDAD DESDE EL DISEÑO 3

2.2 VALIDACIÓN DE ENTRADAS 3

2.3 PROTECCIÓN DE LA INFORMACIÓN 3

2.4 GESTIÓN SEGURA DE ERRORES 3

2.5 PRINCIPIO DE MÍNIMO PRIVILEGIO 3

2.6 USO SEGURO DE BIBLIOTECAS Y DEPENDENCIAS 4

2.7 CALIDAD Y MANTENIBILIDAD DEL CÓDIGO 4

2.8 REVISIÓN DEL CÓDIGO 4

2.9 CUMPLIMIENTO DE LAS DIRECTRICES ESPECÍFICAS 4

3. ORGANIZACIÓN Y USO DE LOS ANEXOS 5

4. CUMPLIMIENTO DE LA GUÍA 6

1. INTRODUCCIÓN

La seguridad del software constituye un elemento esencial para garantizar la confidencialidad, integridad y disponibilidad de la información tratada por las aplicaciones desarrolladas. Un elevado porcentaje de las vulnerabilidades detectadas en aplicaciones software tiene su origen en errores de implementación que podrían evitarse mediante la adopción de prácticas adecuadas durante la fase de desarrollo.

La codificación segura tiene como finalidad reducir la probabilidad de introducir vulnerabilidades en el código fuente mediante la aplicación sistemática de buenas prácticas de programación, el uso adecuado de los mecanismos proporcionados por cada lenguaje y la prevención de patrones de implementación inseguros.

Esta guía establece el marco general de buenas prácticas de codificación segura aplicable a los desarrollos software. Asimismo, define los principios comunes que deberán observarse con independencia del lenguaje de programación empleado.

Las recomendaciones específicas de implementación se desarrollan en los anexos de la presente guía, donde se recogen las particularidades de cada lenguaje de programación utilizado.

1.1 OBJETIVO

El objetivo de esta guía es establecer un conjunto de principios, criterios y recomendaciones generales que permitan desarrollar software de forma segura, reduciendo la probabilidad de introducir vulnerabilidades durante la implementación del código fuente.

La guía proporciona un marco común para la aplicación de prácticas de codificación segura, fomentando el desarrollo de aplicaciones robustas, mantenibles y resistentes frente a amenazas conocidas. Para ello, promueve la adopción de medidas preventivas que permitan minimizar errores de programación, fortalecer la calidad del software y facilitar la detección temprana de defectos relacionados con la seguridad.

Asimismo, este documento pretende homogeneizar los criterios de desarrollo, definiendo una base común sobre la que se apoyan las recomendaciones específicas de cada lenguaje de programación.

Las directrices particulares para JavaScript, Rust, Ada, C y C++ se encuentran en los anexos correspondientes, donde se describen las técnicas, patrones y recomendaciones adaptadas a las características propias de cada tecnología.

1.2 ALCANCE

La presente guía es de aplicación a todos los desarrollos software realizados, mantenidos o evolucionados, independientemente del lenguaje de programación empleado, del tipo de aplicación desarrollada o de la fase del ciclo de vida del software.

Las recomendaciones aquí descritas deberán considerarse durante las actividades de diseño detallado, implementación, revisión de código, pruebas y mantenimiento del software, con el fin de reducir los riesgos asociados a vulnerabilidades derivadas de errores de programación.

Esta guía resulta de aplicación tanto para desarrollos de nueva creación como para la evolución o mantenimiento de aplicaciones existentes, siempre que se realicen modificaciones sobre el código fuente.

El presente documento establece únicamente los principios generales de codificación segura. Las recomendaciones específicas para cada lenguaje de programación se desarrollan en los anexos, los cuales forman parte integrante de esta guía y deberán aplicarse cuando el desarrollo se realice utilizando el lenguaje correspondiente.

La guía complementa las políticas de seguridad, normas de desarrollo, procedimientos de ingeniería del software y procesos de aseguramiento de la calidad establecidos por la organización, sin sustituirlos.

2. PRINCIPIOS GENERALES DE CODIFICACIÓN SEGURA

Los siguientes principios deberán observarse durante el desarrollo de cualquier componente software, independientemente del lenguaje de programación utilizado.

2.1 SEGURIDAD DESDE EL DISEÑO

La seguridad deberá considerarse desde las primeras fases del desarrollo y mantenerse durante todo el ciclo de vida del software. Las decisiones de diseño deberán minimizar la superficie de ataque y reducir la probabilidad de explotación de posibles vulnerabilidades.

2.2 VALIDACIÓN DE ENTRADAS

Toda información procedente de usuarios, aplicaciones externas, sistemas de terceros, ficheros, bases de datos o redes deberá considerarse potencialmente no confiable hasta haber sido validada.

La validación deberá comprobar tanto el formato como el tipo, longitud, rango y contenido esperado de los datos.

2.3 PROTECCIÓN DE LA INFORMACIÓN

Los datos sensibles deberán protegerse adecuadamente durante su almacenamiento, procesamiento y transmisión.

No deberán almacenarse credenciales, claves criptográficas, tokens u otra información confidencial directamente en el código fuente.

2.4 GESTIÓN SEGURA DE ERRORES

Los errores y excepciones deberán gestionarse de forma controlada, evitando revelar información que pueda facilitar un ataque, como rutas internas, consultas SQL, configuraciones del sistema o detalles de implementación.

Los registros generados deberán contener la información necesaria para facilitar el análisis de incidencias sin comprometer la seguridad del sistema.

2.5 PRINCIPIO DE MÍNIMO PRIVILEGIO

Las aplicaciones deberán ejecutarse utilizando únicamente los privilegios estrictamente necesarios para realizar sus funciones.

Deberá evitarse el uso de permisos elevados cuando existan alternativas más restrictivas.

2.6 USO SEGURO DE BIBLIOTECAS Y DEPENDENCIAS

Las bibliotecas, *frameworks* y componentes de terceros deberán proceder de fuentes confiables y mantenerse actualizados.

Antes de incorporar nuevas dependencias deberá evaluarse su estado de mantenimiento, compatibilidad y posibles vulnerabilidades conocidas.

2.7 CALIDAD Y MANTENIBILIDAD DEL CÓDIGO

El código deberá ser claro, consistente y fácilmente comprensible para facilitar su mantenimiento y reducir la probabilidad de introducir errores durante futuras modificaciones.

Se favorecerán soluciones sencillas frente a implementaciones innecesariamente complejas.

2.8 REVISIÓN DEL CÓDIGO

Siempre que sea posible, el código deberá someterse a revisiones por pares y a herramientas de análisis estático y/o dinámico que permitan detectar errores de programación y posibles vulnerabilidades antes de su integración.

2.9 CUMPLIMIENTO DE LAS DIRECTRICES ESPECÍFICAS

Además de los principios recogidos en esta guía, todo desarrollo deberá cumplir las recomendaciones específicas definidas para el lenguaje de programación empleado, según se establece en los anexos correspondientes.

3. ORGANIZACIÓN Y USO DE LOS ANEXOS

Los anexos de la presente guía forman parte integrante del documento y contienen las recomendaciones específicas de codificación segura para lenguajes de programación concretos:

- JavaScript
- Rust
- Ada
- C
- C++

Cada anexo desarrolla las buenas prácticas, patrones de implementación, vulnerabilidades habituales y recomendaciones técnicas propias del lenguaje correspondiente, complementando los principios generales definidos en esta guía.

Los principios recogidos en el cuerpo principal son de aplicación a todos los desarrollos, mientras que las directrices contenidas en cada anexo deberán aplicarse cuando el software se implemente utilizando el lenguaje correspondiente.

4. CUMPLIMIENTO DE LA GUÍA

El cumplimiento de las directrices establecidas en la presente guía tiene como finalidad fomentar el desarrollo de software seguro y reducir la probabilidad de introducir vulnerabilidades durante la implementación del código.

Los principios generales descritos en este documento, junto con las recomendaciones específicas recogidas en los anexos correspondientes a cada lenguaje de programación, deberán ser considerados durante el desarrollo, mantenimiento y evolución de las aplicaciones, adaptándose a las características y necesidades de cada proyecto.

Siempre que sea técnicamente viable, las recomendaciones de codificación segura deberán aplicarse tanto a los desarrollos de nueva creación como a las modificaciones realizadas sobre aplicaciones existentes, priorizando aquellas actuaciones que contribuyan a reducir los riesgos de seguridad identificados.

Cuando, por motivos técnicos, de compatibilidad o de requisitos funcionales, no sea posible aplicar alguna de las recomendaciones establecidas en esta guía o en sus anexos, la desviación deberá estar debidamente justificada y documentada. En estos casos, se recomienda evaluar el riesgo asociado e implantar, cuando sea posible, medidas compensatorias que mitiguen el impacto derivado de la excepción.

La verificación del cumplimiento podrá realizarse mediante los mecanismos de aseguramiento de la calidad definidos por la organización, incluyendo, entre otros:

- Revisiones de código.
- Análisis estático del código fuente.
- Análisis dinámico cuando resulte aplicable.
- Pruebas de seguridad durante el proceso de desarrollo.
- Auditorías técnicas o revisiones periódicas de las aplicaciones.

La presente guía y sus anexos se revisarán periódicamente para mantener su alineación con la evolución de los lenguajes de programación, las amenazas emergentes, las vulnerabilidades conocidas y las buenas prácticas reconocidas por la industria, garantizando así su vigencia y eficacia como referencia para el desarrollo seguro de software.

PUBLICACIONES DE DEFENSA

GOBIERNO DE ESPAÑA
MINISTERIO DE DEFENSA
SUBSECRETARÍA DE DEFENSA
SECRETARÍA GENERAL TÉCNICA
SUBDIRECCIÓN GENERAL DE PUBLICACIONES Y PATRIMONIO CULTURAL