DESARROLLO Y MANTENIMIENTO DE SOFTWARE SEGURO			
PROCESO:	GESTIÓN TICS		
PROCEDIMIENTO:	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
FECHA DE APROBACIÓN:	**CÓDIGO:**	**VERSIÓN:**	**PÁGINA:**
30/10/2025	GTI07-POL01	09	**1** de **13**

POLÍTICA DE DESARROLLO Y MANTENIMIENTO DE SOFTWARE SEGURO

U.A.E CONTADURÍA GENERAL DE LA NACIÓN

GRUPO INTERNO DE TRABAJO DE APOYO INFORMÁTICO

OCTUBRE, 2025

DESARROLLO Y MANTENIMIENTO DE SOFTWARE SEGURO			
PROCESO:	GESTIÓN TICS		
PROCEDIMIENTO:	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
FECHA DE APROBACIÓN:	**CÓDIGO:**	**VERSIÓN:**	**PÁGINA:**
30/10/2025	GTI07-POL01	09	**2** de **13**

CONTROL DE VERSIONES					
Versión	**Sección**	**Tipo**	**Fecha (dd/mm/aa)**	**Autor**	**Observaciones**
0	Todas	Creación	12/11/2010	Giovanny Beltrán	Versión inicial
1.0	Todas	Revisión	21/12/2010	Alberto Torres	Revisión
2.0	Todas	Aprobación	23/12/2010	CARCC	Aprobación
3.0	Todas	Ajustes	27/12/2010	Giovanny Beltrán	Versión Final
4.0	Todas	Ajustes	07/06/2011	Giovanny Beltrán	Actualización
5.0	Todas	Ajustes	11/09/2012	Mónica Merlano.	Revisión
6.0	Todas	Ajustes	16/07/2018	Pedro Grismaldo	Actualización
7.0	Todas	Ajustes	28/11/2018	Grupo Desarrollo	Actualización
8.0	Todas	Ajustes	17/12/2019	Grupo Desarrollo	Actualización
9.0	Todas	Actualización	17/10/2025	Equipo de Sistemas de Información e Información	Actualización del documento

DESARROLLO Y MANTENIMIENTO DE SOFTWARE SEGURO			
PROCESO:	GESTIÓN TICS		
PROCEDIMIENTO:	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
FECHA DE APROBACIÓN:	**CÓDIGO:**	**VERSIÓN:**	**PÁGINA:**
30/10/2025	GTI07-POL01	09	**3** de **13**

Contenido

1. ALCANCE 4
3. TERMINOS Y DEFINICIONES 4
4. CICLO DE DESARROLLO 6
4.1. ITERACIÓN 6
4.2. TAREA 6
4.3. CRITERIO DE LISTO 6
5. PRINCIPIOS GENERALES PARA UN DESARROLLO SEGURO 6
5.1. SEGURIDAD POR DISEÑO (SECURITY BY DESIGN) 6
5.2. DISEÑO LIMPIO Y SOSTENIBLE 6
5.3. TÉCNICAS DE PROGRAMACIÓN SEGURAS 6
5.4. ESTANDARIZACIÓN DEL CICLO DE VIDA Y CRITERIOS DE SEGURIDAD 6
5.5. NO CONFIAR EN DATOS DE ENTRADA 7
5.6. PERMISOS MÍNIMOS 7
5.7. ENTORNOS DE DESARROLLO SEPARADOS 7
5.8. REPOSITORIO DE CÓDIGO: 7
5.9. CANALIZACIÓN SEGURA DE COMPILACIÓN E IMPLEMENTACIÓN (CI/CD) 7
5.10. PRUEBAS DE SEGURIDAD CONTINUAS 7
5.11. MONITOREO Y REPORTES CONTINUOS 7
6. ACTORES Y FUNCIONES 7
7. CICLO DE DESARROLLO SEGURO (METODOLOGÍA SCRUM Y OWASP) ... 8
8. DETALLE DE LAS POLÍTICAS ESPECÍFICAS 8
8.1. GENERALES 8
8.2. PLANEACIÓN DEL GRUPO DE DESARROLLO 9
8.3. DESARROLLO 9
8.4. CERTIFICACIÓN Y PRUEBAS 10
8.6. AMBIENTES DE DESARROLLO DE SOFTWARE 12
8.7. DESARROLLO EXTERNO DE SOFTWARE 12
8.8. SOLICITUDES DE REQUERIMIENTOS 13
9. PERIODICIDAD DE EVALUACIÓN Y REVISIÓN 13

DESARROLLO Y MANTENIMIENTO DE SOFTWARE SEGURO			
PROCESO:	GESTIÓN TICS		
PROCEDIMIENTO:	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
FECHA DE APROBACIÓN:	**CÓDIGO:**	**VERSIÓN:**	**PÁGINA:**
30/10/2025	GTI07-POL01	09	**4** de **13**

INTRODUCCIÓN

Este documento establece los lineamientos para el desarrollo y mantenimiento de software, integrando un enfoque de desarrollo seguro que se aplica en la Contaduría General de la Nación. El objetivo es asegurar que la seguridad digital sea diseñada e implementada integralmente en el ciclo de vida de los sistemas de información, desde la planificación y desarrollo hasta el mantenimiento. Esta política incorpora la metodología Scrum para la gestión ágil y las directrices de seguridad de OWASP (Open Web Application Security Project) para garantizar la calidad del software seguro y la protección contra el uso inapropiado de la información y la mitigación de vulnerabilidades.

1. ALCANCE

Esta política se aplica a todo el ciclo de vida del software desarrollado, adquirido y gestionado por la Contaduría General de la Nación (CGN), incluyendo sistemas nuevos y mejoras a los existentes. Es de cumplimiento obligatorio para todo el personal involucrado, sean funcionarios, contratistas o personal externo, en la operación y mantenimiento de los sistemas. Abarca desde el estudio de viabilidad funcional de la solicitud, ya sea una incidencia o un nuevo requerimiento, hasta la liberación de la versión a producción.

3. TERMINOS Y DEFINICIONES

CISO (Chief Information Security Officer): Responsable de la seguridad general de los sistemas de información de una organización.

Commit: Código asignado para el registro del historial de los ajustes realzados al código de una aplicación de software.

Confidencialidad: Propiedad de la información que consiste en garantizar que esta solo es accedida por personas o sistemas autorizados.

Desarrollo Seguro (SSDLC - Secure System Development Life Cycle): Requisito para generar un servicio, arquitectura, software y sistema seguro, desde la perspectiva del resguardo de la información. Integra los requisitos de seguridad en cada fase del ciclo de vida del desarrollo, ya sea tradicional o ágil.

Disponibilidad: Propiedad de la información que consiste en garantizar el acceso y uso de la información y los sistemas de tratamiento de esta a los individuos, entidades o procesos autorizados cuando lo requieran.

Integridad: Propiedad de la información que consiste en mantener con exactitud la información tal cual fue generada, sin ser manipulada ni alterada por personas o procesos no autorizados.

Iteración: Un ciclo de trabajo durante el cual el equipo de desarrollo estima, planea, se compromete, desarrolla y presenta una versión de software.

Log: Registro oficial de eventos que se presentan durante el desarrollo de un software, que puede contener, fecha, autor, actividad, dirección IP, unidades afectadas, entre otra información.

Mesa de servicio: Punto único de contacto entre los usuarios y el GIT de Apoyo Informático, encargado de gestionar solicitudes, incidentes, problemas y consultas relacionadas con los servicios de TI. Su función principal es gestionar para restaurar el funcionamiento normal de los servicios tecnológicos lo más rápido posible y asegurar una atención eficiente, trazable y alineada con los objetivos institucionales.

OWASP (Open Web Application Security Project): Comunidad abierta dedicada a habilitar a las organizaciones para desarrollar, adquirir y mantener aplicaciones confiables. Sus guías (como OWASP Top 10 y OWASP Cheat Sheet) son referencia para la seguridad de aplicaciones web.

Product Owner: Persona responsable de maximizar el valor, tanto del producto, como del trabajo realizado por el equipo de desarrollo.

Scrum: Marco de trabajo ágil para desarrollar, entregar y mantener productos complejos.

Vulnerabilidad: Condición de debilidad o fragilidad en un activo o control que puede ser explotada por una amenaza, posibilitando un incidente que afecte a la información.

4. CICLO DE DESARROLLO

4.1. ITERACIÓN

Una iteración es un ciclo de trabajo de 2 semanas, tiempo que puede variar y durante el cual el equipo de desarrollo estima, planea, se compromete, desarrolla y entregan funcionalidades o correcciones.

4.2. TAREA

Unidad de trabajo dentro de la iteración, limitada a 4-8 horas.

4.3. CRITERIO DE LISTO

Conjunto de condiciones objetivas que deben cumplirse para aceptar el desarrollo en producción (funcionalidad operativa, pruebas exitosas, documentación completa, etc.).

5. PRINCIPIOS GENERALES PARA UN DESARROLLO SEGURO

La Contaduría General de la Nación, adopta un enfoque de Desarrollo Seguro incorporando requisitos de seguridad en fases del ciclo de vida del proyecto.
Los siguientes principios guiarán el desarrollo de software:

5.1. SEGURIDAD POR DISEÑO (SECURITY BY DESIGN)

La seguridad se integra desde el inicio del ciclo de vida de desarrollo (SSDLC) y se considera en todas sus dimensiones: diseño, desarrollo, pruebas, construcción, fabricación e implementación.

5.2. DISEÑO LIMPIO Y SOSTENIBLE

Se priorizarán diseños sencillos y sostenibles para software y hardware, utilizando arquitecturas conocidas y buenas prácticas, ya que la complejidad es enemiga de la seguridad.

5.3. TÉCNICAS DE PROGRAMACIÓN SEGURAS

Se utilizarán técnicas de programación seguras y buenas prácticas reconocidas, como OWASP, para nuevos desarrollos y en la reutilización de código.

5.4. ESTANDARIZACIÓN DEL CICLO DE VIDA Y CRITERIOS DE SEGURIDAD

Se estandarizarán las actividades, criterios y puntos de control y revisión en cada fase del proceso, incluyendo criterios de seguridad y calidad.

5.5. NO CONFIAR EN DATOS DE ENTRADA
Todos los datos que ingresan a la aplicación deben ser verificados exhaustivamente para garantizar que son los esperados y para evitar ataques por inyección de código.

5.6. PERMISOS MÍNIMOS
Se partirá siempre de un modelo de permisos mínimos, escalando privilegios por demanda de acuerdo con los perfiles establecidos.

5.7. ENTORNOS DE DESARROLLO SEPARADOS
Los entornos de desarrollo, prueba y producción deben ser independientes, y lo más similares posible en cuanto a controles de seguridad.

5.8. REPOSITORIO DE CÓDIGO:
El repositorio de documentación y código fuente debe estar protegido por credenciales corporativas, con registro de cambios y copias de seguridad automáticas.

5.9. CANALIZACIÓN SEGURA DE COMPILACIÓN E IMPLEMENTACIÓN (CI/CD)
Se adoptará la integración continua como procedimiento estándar, con pruebas automáticas de seguridad en cada commit/versión para garantizar el cumplimiento de los requisitos de seguridad y calidad.

5.10. PRUEBAS DE SEGURIDAD CONTINUAS
Se realizarán pruebas de seguridad estáticas, dinámicas y de intrusión.

5.11. MONITOREO Y REPORTES CONTINUOS
Se generarán informes y abordarán los puntos débiles detectados con planes de corrección o mitigación.

5.12. AGILIDAD CON SCRUN
Iteraciones cortas, entregas incrementales, roles definidos y tableros de seguimiento.

6. ACTORES Y FUNCIONES
Los actores y funciones de cada uno lo encontramos detallado en la metodología de desarrollo de software y basado en la metodología de desarrollo ágil SCRUM. Para más información consultar el documento nombrado.

7. CICLO DE DESARROLLO SEGURO (METODOLOGÍA SCRUM Y OWASP)

El ciclo de desarrollo de software se encuentra detallado en la metodología de desarrollo de software y basado en la metodología de desarrollo ágil SCRUM. Para más información consultar el documento nombrado.

Las prácticas definidas por OWASP (Open Web Application Security Project) constituyen un marco esencial para desarrollar, mantener y operar software seguro. En especial se adopta la categoría riesgos críticos de seguridad en aplicaciones web. Como buenas prácticas que se integran en los desarrollos de software en la CGN.

8. DETALLE DE LAS POLÍTICAS ESPECÍFICAS

Esta sección detalla las políticas operativas, actualizadas para reflejar el enfoque de desarrollo seguro.

8.1. GENERALES

- Toda solicitud debe realizarse a través de la Mesa de Servicio y debe documentarse anexando los documentos que soporten dicha solicitud.
- Todo incidente generado por desarrollo de software y solicitudes de requerimientos de software posterior a su aprobación serán tratadas y gestionadas en la plataforma de seguimiento y gestión de proyectos JIRA durante el ciclo de desarrollo.
- Antes de aprobar una nueva solicitud, esta debe contar con todos los estudios de impacto técnico, funcional y operativo, incluyendo una valoración de riesgos de seguridad digital.
- Toda especificación de desarrollo de software debe tener definidas las pruebas de aceptación por parte del analista funcional.
- La liberación de una nueva versión de software se realiza en conjunto con el Grupo de Certificación (QA) y el product owner (PO), teniendo en cuenta la complejidad, urgencia y número de nuevos requerimientos y se procede a la realización de una gestión de cambio.
- Únicamente se ponen en producción desarrollos y funcionalidades con Acta de recibo a satisfacción del analista funcional y la aprobación del Grupo de Certificación que avale la calidad del desarrollo y la seguridad.
- La documentación referente al ciclo de vida del software debe quedar en el repositorio de documentos dispuesto para la entidad para tal fin.

8.2. PLANEACIÓN DEL GRUPO DE DESARROLLO

- Las planeaciones de cada iteración (Sprint) deben tener en cuenta un porcentaje de tiempo para las actividades adicionales de soporte y mantenimiento, coordinado con el líder técnico.
- Todo requerimiento que hace parte de la planeación para ser desarrollado debe ser segmentado en múltiples tareas con el propósito de r educir la complejidad. Toda tarea debe tener una estimación de duración entre 4 horas y 8 horas de esfuerzo como máximo.
- Cada semana como mínimo se debe realiza una reunión informativa (Daily Scrum) donde cada integrante del equipo expone su avance y problemas durante la iteración (Sprint).
- Cualquier actividad adicional no planeada urgente que altere la planeación inicial de la iteración debe ser nuevamente valorada y generará un nuevo plan de la iteración.
- Al final de cada iteración (Sprint) se construye una versión para pruebas de certificación, incluyendo la verificación de seguridad, según sea necesario.

8.3. DESARROLLO

- Todo desarrollo de software deberá aplicar los lineamientos definidos en las historias de usuario y aquellos que el líder técnico considere necesarios basados enOWASP.
- Para cuando se desarrollen pruebas unitarias, cualquier prueba unitaria que falle debe ser solucionada de inmediato. Estas pruebas unitarias deben incluir casos que validen aspectos de seguridad.
- Al realizar un "commit" en el sistema de versionamiento (ej. Subversion o Git), se deberá incluir una descripción clara de las modificaciones realizadas. De la misma forma debe tenerse en cuenta la guía patrón de nombramiento documentos de desarrollo de software. Este registro debe incluir la persona que ha realizado cada cambio y cuándo, y toda actividad se debe documentar.
- Se establece como registro válido para reportar las entregas de los desarrollos a pruebas de certificación, el formulario en línea que provee el equipo de certificación de softwareSe deben validar todas las entradas de datos (tipo, rango, longitud, caracteres especiales, valores nulos) y sanear las salidas para prevenir inyecciones SQL, XSS y otras vulnerabilidades.

- Se deben implementar controles de autenticación robustos (hashes para contraseñas, bloqueo de cuentas, CAPTCHA) y una gestión de sesiones segura (expiración, cierre de sesión, no exponer identificadores de sesión).
- Los mensajes de error no deben revelar información sensible sobre la tecnología o la aplicación; deben ser genéricos.
- Se debe garantizar que el código fuente no pueda ser descargado ni modificado por usuarios no autorizados, y no se deben ejecutar comandos directamente en el sistema operativo desde la aplicación.
- Se debe garantizar la generación y almacenamiento de logs de auditoría sobre tablas y transacciones críticas, incluyendo ID de usuario, fecha, hora, acción, tabla modificada y valores.
- Las credenciales secretas (ej. contraseñas, claves privadas) se aislarán lógicamente de la base del código principal y no se permitirán credenciales de codificación fija (hardcodeadas) en el código fuente.
- Los marcos y bibliotecas de terceros deben ser probados y aprobados antes de su uso.

8.4. CERTIFICACIÓN Y PRUEBAS

- Para que un desarrollo de software pueda ser sometido a pruebas, el equipo de desarrollo debe suministrar debidamente organizados los siguientes documentos, según corresponda.
    - Solicitud requerimiento en la plataforma GLPI
    - Historia de Usuario.
    - Guiones de prueba de software, y de seguridad.
- Las solicitudes de desarrollo de software aprobadas deberán cumplir exitosamente las Pruebas de Certificación (incluyendo pruebas básicas, pruebas obligatorias y de ser necesario pruebas críticas como estrés, carga, integridad y seguridad) para poder ser puestas en producción.
- Para que un desarrollo sea puesto en producción es necesario contar con el acta de aceptación de usuario firmada por todos los responsables, firmada por el grupo de certificación que avale la calidad del desarrollo y el aval del especialista en ciclo de vida de desarrollo seguro que certifique la seguridad, junto con el acta de certificación de versión.
- Mantener un control de versiones de todos los despliegues, se debe permitir recuperar versiones específicas.
- Los cambios importantes ejecuten un proceso formal de documentación, especificación, pruebas, control de calidad y gestión de la implementación.

- Todos los cambios deben ser probados y documentados completamente, y se requerirá el escaneo de vulnerabilidades y un plan para la mitigación de estas.
- Cada vez que se solucionan incidencias en una versión que se encuentran en proceso de certificación, se debe construir una nueva versión para realizar nuevamente el proceso de certificación.
- Control de Versiones de Código fuente: En la etapa de versionamiento se debe verificar que en el Sistema de Control de versiones (SVN/Git), la etiqueta (TAG) de producción solo tenga las revisiones "commits" relacionados con las solicitudes de cambios registradas. Se debe verificar que la versión de certificación que se va a mezclar con el tronco del árbol de desarrollo contenga únicamente los desarrollos que conforman la versión de producción. Se debe definir el número de la versión y establecer los desarrollos agregados, junto con el historial de cambios de último momento, los cuales se registran en el acta de Certificación de versión.
- Se realizarán pruebas de seguridad funcional, incluyendo autenticación, complejidad de contraseñas, bloqueo de cuentas, funcionalidad de captchas, revisión de logs y mensajes de error.
- Se realizarán análisis estáticos y dinámicos del código (SAST y DAST) para identificar vulnerabilidades. Las herramientas automáticas se preferirán cuando sea posible.
- Se realizarán pruebas de intrusión (Ethical Hacking), en un ambiente productivo definitivo, pero con acceso controlado.

8.5. DATOS DE PRUEBA

Protección de los datos de producción en pruebas: Para las pruebas de las aplicaciones que lo requieran, se utilizarán datos extraídos del ambiente de producción los cuales deben ser protegidos mediante el cumplimiento de la totalidad de los siguientes requisitos:

- Se debe solicitar realizar una copia de las bases de datos de producción como base de prueba
- Las copias de las bases de datos de producción deberán ser alteradas con datos ficticios o anonimizadas, específicamente las que contengan datos personales sensibles y/o privados.
- Se aplicarán los mismos procedimientos de control de acceso que a los sistemas operacionales a los sistemas de información de pruebas.

DESARROLLO Y MANTENIMIENTO DE SOFTWARE SEGURO			
PROCESO:	GESTIÓN TICS		
PROCEDIMIENTO:	DESARROLLO Y MANTENIMIENTO DE SOFTWARE		
FECHA DE APROBACIÓN:	**CÓDIGO:**	**VERSIÓN:**	**PÁGINA:**
30/10/2025	GTI07-POL01	09	**12** de **13**

8.6. AMBIENTES DE DESARROLLO DE SOFTWARE

Para reducir la probabilidad de alteración de programas fuente, se aplicarán las siguientes políticas:

1. La Administración de las distintas versiones de un software será gestionada mediante una herramienta de software especializada para el control de versiones.
2. Se restringirá el acceso a todo operador y/o usuario que no pertenezca al área de desarrollo de aplicaciones a los ambientes de desarrollo y a las herramientas que permitan la generación y/o manipulación de código fuente.
3. Se realizarán copias de Respaldo del código fuente según los procedimientos definidos.
4. Los ambientes de desarrollo, pruebas, capacitación, contingencia y producción (los que tenga el sistema o software) estarán separados, preferentemente en forma física o lógica según se considere.
5. Se verificará que los aplicativos cuenten con las últimas versiones estables y parches de seguridad de software, sistema operativo, servidor de aplicaciones, base de datos, máquina virtual, etc., antes del despliegue.
6. Se controlará que los usuarios no tengan acceso a archivos de configuración o directorios sensibles; solo usuarios con privilegios o autorizados deberán tener acceso.
7. Los usuarios de la aplicación y el software que se ejecuten en el servidor (base de datos, sftp, apache, iis, jboss, tomcat, etc.) deben tener los mínimos privilegios sobre el sistema.
8. Se utilizarán protocolos seguros (SSH, SFTP, FTPS, VPN SSL, IP SEC) para la comunicación y transferencia de archivos.

8.7. DESARROLLO EXTERNO DE SOFTWARE

Para el caso que se considere la tercerización del desarrollo de software, se establecerán las siguientes políticas:

1. Se debe establecer con el proveedor externo y con el mayor detalle el alcance de los Acuerdos de licencias, propiedad de código, derechos de Propiedad Intelectual y Patrimonial, ajustados siempre al cumplimiento del marco de la legislación.
2. Definir los Requerimientos contractuales con respecto a la calidad del código y la existencia de garantías.

3. Especificar los Procedimientos de certificación de la calidad y precisión del trabajo llevado a cabo por el proveedor, que incluyan auditorías, revisión de código para detectar código malicioso, verificación del cumplimiento de los requerimientos de seguridad del software establecido y la entrega de evidencias de pruebas suficientes para garantizar la protección contra contenido malicioso y vulnerabilidades conocidas (ej. OWASP Top 10).
4. Cumplir las políticas de seguridad de la información de la Contaduría General de la Nación
    - Los contratos con empresas proveedoras deben contener cláusulas que aseguren los niveles de confidencialidad de la información y exijan el cumplimiento de los requisitos de seguridad.
5. Se debe supervisar y hacer seguimiento a la actividad de desarrollo de sistemas contratados externamente, verificando el cumplimiento de prácticas seguras de diseño, codificación y pruebas.
6. Se realizará un proceso de verificación de código antes de su salida a producción, incluyendo aquellas librerías de terceros.

8.8. SOLICITUDES DE REQUERIMIENTOS

Con el fin de minimizar los riesgos de alteración de los sistemas de información, se aplican los controles establecidos en el Procedimiento de Desarrollo de Software, relacionados con las solicitudes de requerimientos, para lo cual se establecen como políticas lo siguiente:

1. Solo se evaluarán las presentadas por usuarios autorizados.
2. Solo serán aprobadas con el concepto positivo de la evaluación funcional y técnica, incluyendo la valoración de riesgos de seguridad digital y el análisis de impactos en la seguridad.

9. PERIODICIDAD DE EVALUACIÓN Y REVISIÓN

Esta política será evaluada y revisada anualmente, o con una periodicidad máxima de dos años, para asegurar su cumplimiento y la incorporación de todas las normas y procedimientos necesarios en el marco de la Ciberseguridad y Seguridad de la Información.