TABLA CONTENIDO

1. OBJETIVO

Establecer los lineamientos y directrices para el inicio, planificación, desarrollo, pruebas, implementación, mantenimiento y cierre de proyectos de software realizados por **[NOMBRE DE LA EMPRESA]**o por terceros en su nombre, garantizando la integración de prácticas de seguridad de la información en todas las fases del ciclo de vida del software.

2. ALCANCE

Este procedimiento aplica a todas las actividades relacionadas con el desarrollo, adquisición, mantenimiento, mejora o retiro de software y aplicaciones que operen en la infraestructura tecnológica de **[NOMBRE DE LA EMPRESA]**, incluyendo:

- Software desarrollado internamente por el equipo de tecnología o por contratistas bajo su supervisión.
- Software adquirido o desarrollado por terceros cuando sea integrado, configurado o personalizado para la organización.
- Servicios en la nube o componentes de terceros que formen parte del entorno de desarrollo o de producción.
- Procesos de integración y pruebas de aplicaciones, bases de datos, interfaces y APIs.
- Gestión de información compartida con proveedores, aliados estratégicos o partes interesadas en el marco de actividades de desarrollo o mantenimiento.

El alcance abarca a los roles involucrados en el ciclo de vida del desarrollo y mantenimiento de software dentro de la organización, incluyendo: Arquitecto de Soluciones, Director de TI (CIO), Gerente de Proyectos, PMO, Scrum Master / Agile Coach, Líder Técnico (Tech Lead), Desarrolladores (junior, senior y pasantes), Analistas de área, Analista de I+D, Analista de Seguridad de la Información, Coordinación de Calidad y Cumplimiento,

3. REFERENCIAS NORMATIVAS

El presente procedimiento se fundamenta en las siguientes normas y marcos de referencia los cuales establecen principios, controles y prácticas recomendadas para el desarrollo seguro de software:

- **NIST SP 800-218 - Secure Software Development Framework (SSDF):** Este marco define un conjunto de prácticas para integrar la seguridad en el ciclo de vida del desarrollo de software (SDLC). Su aplicación permite incorporar controles de seguridad desde las primeras fases del desarrollo (“**shift-left security**”), reducir la probabilidad de vulnerabilidades en los productos liberados, fortalecer la

protección de los componentes del software frente a accesos no autorizados y promover la trazabilidad y la mejora continua de los procesos de desarrollo.

- **NIST SP 800-30 - Guide for Conducting Risk Assessments:** Proporciona directrices para la identificación, análisis y evaluación de riesgos de seguridad de la información. En el contexto del desarrollo de software, permite estructurar el análisis de amenazas, vulnerabilidades e impactos asociados a los sistemas, facilitando la priorización de controles de seguridad y la toma de decisiones basada en riesgos a lo largo del ciclo de vida del software.
- **NIST SP 800-53 - Security and Privacy Controls for Information Systems and Organizations:** Establece un catálogo integral de controles de seguridad y privacidad organizados en familias (control de acceso, auditoría, gestión de configuraciones, entre otros). Su aplicación en el desarrollo de software permite definir requisitos de seguridad alineados a estándares internacionales, asegurar la implementación de controles técnicos y administrativos, y fortalecer la gestión integral del riesgo en sistemas de información.
- **NIST SP 800-61 - Computer Security Incident Handling Guide:** Proporciona lineamientos para la gestión de incidentes de seguridad de la información. En el desarrollo de software, orienta la definición de procedimientos para la detección, análisis, contención, erradicación y recuperación frente a incidentes relacionados con aplicaciones, así como la generación de lecciones aprendidas que fortalezcan la seguridad en futuras versiones del software.
- **OWASP SAMM - Software Assurance Maturity Mode:** Proporciona lineamientos prácticos y medibles para evaluar la madurez y las brechas de seguridad del programa de desarrollo de software. Este modelo orienta la implementación gradual de prácticas seguras en las áreas de gobernanza, diseño, implementación, verificación y operación.
- **OWASP (Open Web Application Security Project) Top 10:** Identifica las diez vulnerabilidades de seguridad más críticas para aplicaciones web. Se utiliza como guía prioritaria para la definición de requisitos de seguridad, el modelado de amenazas, las listas de verificación (checklists) de código seguro y los criterios de aceptación en las pruebas de seguridad.
- **OWASP Testing Guide:** Proporciona una metodología integral para la ejecución de pruebas de seguridad en aplicaciones web. Incluye técnicas para la identificación de vulnerabilidades mediante pruebas manuales y automatizadas, cubriendo aspectos como autenticación, control de acceso, validación de entradas, configuración segura y lógica de negocio. Su aplicación permite estandarizar las pruebas de seguridad dentro del proceso de aseguramiento de calidad del software.
- **ISO/IEC 27034 - Seguridad en el ciclo de vida del software:** Establece los principios de gestión de seguridad de aplicaciones dentro del Sistema de Gestión de Seguridad de la Información (SGSI). Proporciona un marco que permite integrar

requisitos de seguridad de la información en cada fase del desarrollo, definir roles, responsabilidades y criterios de aceptación segura del software, garantizar la coherencia entre los procesos de desarrollo, la gestión de riesgos y los controles del SGSI y, establecer métricas y evidencias de cumplimiento aplicables en auditorías internas y externas.

- **Reglamento General de Protección de Datos - RGPD (UE 2016/679):** Reglamento europeo establece las obligaciones legales para la protección de los datos personales en el tratamiento realizado por entidades públicas o privadas. En el contexto del desarrollo de software, se aplican los siguientes principios:
    - **Minimización de datos y limitación del tratamiento:** los sistemas deben recolectar y procesar únicamente los datos estrictamente necesarios para la finalidad prevista.
    - **Privacidad desde el diseño y por defecto (Privacy by Design & by Default):** los requisitos de protección de datos deben considerarse desde las etapas de análisis y diseño del software.
    - **Seguridad de los datos personales:** el software debe incluir medidas de cifrado, control de acceso y trazabilidad de la información.
    - **Gestión de derechos del titular:** los sistemas deben permitir ejercer los derechos de acceso, rectificación, supresión y portabilidad de datos.

4. DEFINICIONES

Acción correctiva: Acción para eliminar la causa de una no conformidad (3.6.9) y evitar que vuelva a ocurrir

Amenaza: Causa potencial de un incidente no deseado que puede resultar en daño a un sistema u organización.

Confidencialidad: Propiedad que determina que la información no esté disponible ni sea revelada a individuos, entidades o procesos no autorizados

Datos: Hechos o valores, que pueden ser estructurados o no estructurados, susceptibles de ser procesados y que, al ser interpretados, se convierten en Información.

Desarrollo Seguro de Software (SSD): El proceso de integrar actividades y controles de seguridad (como el modelado de amenazas, revisión de código seguro, y pruebas de seguridad) en todas las fases del ciclo de vida del desarrollo (SDLC).

Disponibilidad: Propiedad de que la información sea accesible y utilizable por solicitud de una entidad autorizada.

Eficacia: Grado en el que se realizan las actividades planificadas y se logran los resultados planificados.

Eficiencia. Relación entre el resultado alcanzado y los recursos utilizados.

Especificación: Documento que establece requisitos

Evento de seguridad de la información: presencia identificada de una condición de un sistema, servicio o red, que indica una posible violación de la política de seguridad de la información o la falla de las salvaguardas, o una situación desconocida previamente que puede ser pertinente a la seguridad.

Incidente de seguridad de la información: un evento o serie de eventos de seguridad de la información no deseados o inesperados, que tienen una probabilidad significativa de comprometer las operaciones del negocio y amenazar la seguridad de la información.

Información. Datos procesados o interpretados que poseen significado, contexto y relevancia, y que son esenciales para la operación del negocio.

Inspección. Determinación de la conformidad con los requisitos especificados.

Integridad: Propiedad de salvaguardar la exactitud y completitud de los activos. En el desarrollo, garantiza que el código, los datos y los binarios no hayan sido modificados de forma no autorizada.

Modelado de Amenazas: Proceso estructurado de identificación, comunicación y comprensión de amenazas y mitigaciones dentro del contexto de un sistema.

No Repudio: Propiedad que asegura que la acción o evento de un usuario o entidad no puede ser negada posteriormente.

Riesgo: Efecto de la incertidumbre sobre los objetivos. Se mide en términos de la probabilidad de ocurrencia de una amenaza y el impacto potencial que causaría en la organización.

Seguimiento. Determinación del estado de un sistema, un proceso, un producto, un servicio o una actividad

Seguridad de la información preservación de la confidencialidad, la integridad y la disponibilidad de la información; además, puede involucrar otras propiedades tales como: autenticidad, trazabilidad (Accountability), no repudio y fiabilidad

Sistema de información: Conjunto de aplicaciones, hardware, software, redes y recursos humanos que recolectan, procesan, almacenan y distribuyen información.

Verificación: Confirmación, mediante la aportación de evidencia objetiva de que se han cumplido los requisitos especificados.

Vulnerabilidad: Debilidad de un activo o control que podría ser explotada por una o más amenazas.

5. RESPONSABILIDADES

Alta dirección (Director de TI / CIO y Representante Legal)

- Proveer los recursos humanos, técnicos y financieros necesarios para la ejecución de las actividades de desarrollo y mantenimiento de software.
- Definir la estrategia tecnológica y asegurar su alineación con los objetivos del negocio.
- Aprobar políticas, lineamientos técnicos y de seguridad de la información.
- Establecer instancias de gobierno, priorización y seguimiento de proyectos (PMO, comités).
- Supervisar el cumplimiento del SGSI en los proyectos de software.
- Garantizar la adecuada gestión de riesgos, cumplimiento normativo y protección de la información.
- Definir el apetito de riesgo organizacional y asegurar su adecuada gestión en los proyectos TI.
- Promover una cultura organizacional orientada a la seguridad de la información y mejora continua.
- Asegurar la asignación de responsabilidades claras en materia de seguridad y desarrollo de software.
- Evaluar periódicamente el desempeño del SGSI mediante indicadores estratégicos y auditorías.

Gerente de Proyectos / PMO

- Planificar, coordinar y hacer seguimiento a los proyectos de desarrollo de software.
- Asegurar la integración de controles de seguridad en todas las fases del proyecto.
- Gestionar cronogramas, recursos, riesgos y partes interesadas.
- Definir criterios de aceptación y asegurar el cumplimiento de entregables.
- Consolidar reportes de avance, indicadores y lecciones aprendidas.
- Velar por la correcta documentación del proyecto conforme al SGSI.
- Asegurar la gestión formal de cambios (control de cambios) durante el ciclo de vida del proyecto.
- Monitorear riesgos del proyecto y coordinar planes de tratamiento en conjunto con seguridad de la información.
- Garantizar la trazabilidad entre requerimientos, desarrollo, pruebas y liberación.
- Alinear la ejecución del proyecto con metodologías definidas (ágiles, híbridas o tradicionales).

Scrum Master / Agile Coach

- Facilitar la implementación de marcos ágiles asegurando la incorporación de prácticas de seguridad.
- Eliminar impedimentos que afecten el desarrollo seguro del software.
- Promover la mejora continua, retrospectivas y buenas prácticas del equipo.
- Asegurar que los eventos ágiles integren criterios de calidad y seguridad.

- Fomentar la integración de prácticas DevSecOps dentro del ciclo ágil.
- Asegurar la visibilidad de riesgos y vulnerabilidades dentro del backlog.
- Promover la autoorganización del equipo bajo principios de responsabilidad compartida en seguridad.

Arquitecto de Soluciones

- Diseñar la arquitectura de software considerando seguridad, escalabilidad y rendimiento.
- Incorporar principios de seguridad por diseño (OWASP SAMM, NIST SSDF, ISO/IEC 27034).
- Validar decisiones tecnológicas y patrones de diseño.
- Revisar y aprobar la documentación técnica.
- Asesorar en la mitigación de riesgos técnicos y vulnerabilidades.
- Definir estándares tecnológicos, frameworks y lineamientos de desarrollo.
- Evaluar impactos de nuevas tecnologías en términos de seguridad y cumplimiento.
- Asegurar la interoperabilidad y consistencia entre sistemas y componentes.

Líder Técnico / Tech Lead

- Coordinar técnicamente al equipo de desarrollo.
- Asegurar la correcta implementación de estándares de desarrollo seguro.
- Supervisar revisiones de código y calidad técnica.
- Apoyar la resolución de incidentes técnicos y vulnerabilidades.
- Garantizar la coherencia entre arquitectura y desarrollo.
- Asegurar la correcta implementación de pipelines de integración y despliegue continuo (CI/CD).
- Monitorear la deuda técnica y promover su reducción.
- Validar el cumplimiento de buenas prácticas de versionamiento y control de código fuente.

Analista de Seguridad de la Información

- Definir y validar controles de seguridad aplicables al desarrollo de software.
- Realizar análisis de vulnerabilidades y gestión de riesgos.
- Acompañar pruebas de seguridad (SAST, DAST, IAST).
- Verificar cumplimiento del SGSI en los proyectos.
- Gestionar incidentes de seguridad relacionados con aplicaciones.
- Realizar análisis de amenazas (threat modeling) en etapas tempranas del desarrollo.
- Definir requisitos de seguridad para aplicaciones y servicios.
- Monitorear vulnerabilidades emergentes y coordinar su remediación.

- Apoyar auditorías internas y externas relacionadas con seguridad de la información.

Coordinación de Calidad y Cumplimiento (QA)

- Definir y ejecutar estrategias de aseguramiento de calidad.
- Validar el cumplimiento de requisitos funcionales, técnicos y de seguridad.
- Gestionar pruebas funcionales, de rendimiento y seguridad.
- Asegurar la trazabilidad de pruebas e incidencias.
- Verificar la remediación de defectos y vulnerabilidades.
- Definir criterios de calidad y métricas de aceptación del software.
- Asegurar la automatización de pruebas cuando sea aplicable.
- Validar la preparación del software para su liberación (go-live).

Desarrolladores (Junior, Senior, Pasantes)

- Aplicar prácticas de desarrollo seguro (OWASP Top 10 y guías internas).
- Participar en revisiones de código y pruebas técnicas.
- Documentar funcionalidades, cambios y dependencias.
- Reportar vulnerabilidades, errores o fallos detectados.
- Cumplir lineamientos del SGSI, confidencialidad y protección de datos.
- Implementar controles de seguridad en el código (validaciones, manejo de errores, autenticación, etc.).
- Asegurar el uso adecuado de librerías y dependencias seguras.
- Participar en actividades de integración continua y despliegue.
- Mantener la calidad del código mediante buenas prácticas y estándares definidos.

Analistas de área / Analista I+D

- Levantar y documentar requerimientos funcionales y no funcionales.
- Asegurar la inclusión de requisitos de seguridad desde el diseño.
- Validar que las soluciones respondan a necesidades del negocio.
- Apoyar procesos de innovación y mejora continua.
- Definir criterios de aceptación claros y verificables.
- Asegurar la trazabilidad entre necesidades del negocio y entregables técnicos.
- Participar en validaciones funcionales con usuarios finales.
- Identificar oportunidades de mejora en procesos y soluciones tecnológicas.

Responsable de Seguridad de la Información

- Asesorar a los equipos de tecnología y proyectos en la implementación de prácticas de desarrollo seguro y en la gestión de riesgos asociados al software.

- Validar el cumplimiento de las políticas del SGSI y de los controles definidos conforme a ISO/IEC 27001 e ISO/IEC 27034.
- Definir lineamientos, estándares y requisitos de seguridad aplicables al ciclo de vida del desarrollo de software.
- Apoyar la ejecución de revisiones de seguridad, análisis de vulnerabilidades y pruebas de penetración (internas o externas).
- Consolidar, reportar y hacer seguimiento a hallazgos de seguridad, coordinando acciones de tratamiento con las áreas responsables.
- Asegurar la integración de la seguridad en el ciclo de vida del desarrollo (enfoque DevSecOps).
- Monitorear el estado de riesgos de seguridad de la información y reportar a la alta dirección.
- Apoyar auditorías internas y externas del SGSI y procesos de mejora continua.

Trabajadores (Usuarios internos de la organización)

- Reportar oportunamente incidentes, eventos de seguridad o vulnerabilidades identificadas en el uso de los sistemas de información.
- Cumplir con las políticas, normas y procedimientos del SGSI establecidos por la organización.
- Hacer uso adecuado de los activos de información, garantizando la confidencialidad, integridad y disponibilidad de los datos.
- Mantener la confidencialidad de la información y el correcto uso de credenciales de acceso.
- Participar en actividades de capacitación, sensibilización y pruebas cuando su rol lo requiera.
- Adoptar buenas prácticas de seguridad en el uso de herramientas tecnológicas (gestión de contraseñas, manejo de información, prevención de phishing, entre otros).
- Notificar el uso inadecuado o accesos no autorizados a los sistemas de información.

6. PROCEDIMIENTO

El presente procedimiento integra prácticas de desarrollo seguro bajo un enfoque DevSecOps, articulando el Sistema de Gestión de Seguridad de la Información (SGSI) con marcos de referencia internacionales como NIST, OWASP e ISO/IEC 27034, e incorporando metodologías ágiles como Scrum cuando aplique. Esto permite garantizar la integración continua de la seguridad, la gestión de riesgos y la mejora continua a lo largo del ciclo de vida del software.

6.1. Inicio y requisitos

El proceso de desarrollo de software inicia con la identificación de una necesidad tecnológica o funcional dentro de **[NOMBRE DE LA EMPRESA]**, la cual puede ser propuesta por cualquier área usuaria o por el área de Tecnología de la Información (IT).

En los proyectos que adopten marcos ágiles (Scrum), esta fase se articula mediante la definición inicial del **Product Backlog**, donde se incorporan los requerimientos funcionales, no funcionales y de seguridad como historias de usuario. Los criterios de aceptación deberán incluir controles de seguridad y privacidad, en alineación con el enfoque DevSecOps y el marco NIST SP 800-218 (SSDF).

6.1.1. Levantamiento de la solicitud

La solicitud de desarrollo, actualización o mejora de un software debe ser registrada mediante el formato de Solicitud de gestión del cambio. El Gerente de Proyectos / PMO revisará la solicitud para determinar su viabilidad técnica, económica, el cumplimiento de requisitos de privacidad y protección de datos y el nivel de riesgo de seguridad de la información que representa. Se definirá si el desarrollo será interno o tercerizado (contratado con un proveedor). Asimismo, se deberá asegurar que la solicitud contemple los requisitos de seguridad y privacidad desde su origen, en cumplimiento del principio de **"Security and Privacy by Design".** La solicitud deberá quedar registrada como información documentada controlada dentro del SGSI.

6.1.2. Definición de requerimientos

Durante esta fase se documentan los requisitos funcionales y no funcionales del sistema. Asimismo, se asegura que se incluyan los criterios de seguridad, privacidad y cumplimiento normativo, conforme a los siguientes lineamientos:

- **Requisitos funcionales:** describen las funcionalidades esperadas del software, flujos de trabajo, usuarios y roles.

- **Requisitos no funcionales:** incluyen aspectos de rendimiento, disponibilidad, escalabilidad, mantenibilidad, usabilidad y seguridad de la información.

Los requisitos de seguridad deben considerar los siguientes elementos mínimos:

- **Autenticación y control de acceso:** mecanismos basados en roles y privilegios mínimos (principio de mínimo privilegio y segregación de funciones).

- **Protección de datos:** cifrado de información sensible en tránsito y en reposo.

- **Gestión de sesiones y tokens:** tiempos de expiración, revocación y almacenamiento seguro.

- **Validación y saneamiento de entradas:** prevención de inyecciones y ataques comunes (según OWASP Top 10).

- **Gestión de Componentes:** Inclusión de requisitos para el uso de librerías, frameworks y componentes actualizados, libres de vulnerabilidades conocidas.

- **Registro y trazabilidad:** logs de acceso cambios y operaciones relevantes que permitan el no repudio y la auditoría.

- **Gestión de errores y excepciones:** manejo seguro de errores sin exposición de stack traces ni información sensible al usuario final.

- **Privacidad y cumplimiento RGPD:** recopilación mínima de datos personales, consentimiento informado y mecanismos de ejercicio de derechos.

Los requisitos de seguridad deberán alinearse con los controles aplicables del NIST SP 800-53, especialmente en las familias de control de acceso (AC), auditoría (AU), identificación y autenticación (IA) y protección de sistemas (SC). En entornos ágiles, estos requisitos deberán incorporarse como criterios de aceptación de historias de usuario, asegurando su validación en cada iteración (Sprint).

6.1.3. Identificación y análisis de riesgos

El equipo de desarrollo, con apoyo del Responsable de Seguridad de la Información y Analista de Seguridad de la Información, realizará una evaluación formal de riesgos de seguridad y cumplimiento del proyecto.

Se aplicará un Modelo de Amenazas (Threat Modeling), siguiendo enfoques como STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) o similar, para identificar vulnerabilidades y amenazas potenciales desde la fase de diseño.

Amenaza	**Principio de Seguridad**	**Descripción del Ataque**
Spoofing (Suplantación)	Autenticidad	Un atacante se hace pasar por otra persona o entidad.
Tampering (Manipulación)	Integridad	Modificación no autorizada de datos o código.
Repudiation (Repudio)	No Repudio	El atacante niega haber realizado una acción debido a la falta de trazabilidad.
Information **D**isclosure (Divulgación)	Confidencialidad	Exposición de información sensible a personas no autorizadas.
Denial of **S**ervice (Denegación de Servicio)	Disponibilidad	Impedir a usuarios legítimos acceder a los recursos del sistema.

Elevation of Privilege (Elevación de Privilegios)	Autorización	Un usuario obtiene capacidades o accesos superiores a los que le corresponden.

Los riesgos identificados deben documentarse en el registro de riesgos, evaluando su probabilidad e impacto para determinar el nivel de riesgo inherente. Posteriormente, se documentarán los controles de mitigación propuestos, obteniendo el riesgo residual. La metodología de análisis de riesgos deberá alinearse con lo establecido en el NIST SP 800-30, garantizando un enfoque sistemático para la identificación, estimación y tratamiento de riesgos. Los resultados del análisis de riesgos deberán integrarse al backlog del proyecto como **historias técnicas o de seguridad**, priorizadas según su nivel de riesgo.

6.1.4. Clasificación de activo de la información

Para cumplir con el **"Shift-Left Security"** (NIST SSDF), la fase de inicio es donde se debe clasificar el activo de información. Una vez validada la solicitud, el comité de seguridad debe clasificar el sistema propuesto en función de la sensibilidad de los datos que manejará y su impacto en el negocio:

El Gerente de proyecto clasificará el proyecto conforme a su criticidad e impacto sobre los activos de información, considerando:

Nivel de confidencialidad, integridad y disponibilidad de la información tratada.

Tipo de datos procesados (personales, financieros, internos, públicos).

Dependencia operativa del software dentro de los procesos de negocio.

Esta clasificación inicial definirá la línea base del nivel de controles de seguridad que se deben aplicar durante el ciclo de vida del software y se registrará en el Inventario de activos. Esta actividad se articula con el proceso de gestión de activos del SGSI, asegurando la trazabilidad entre el activo de información, los riesgos asociados y los controles implementados.

6.1.5. Aprobación de inicio

- Una vez definidos los requisitos, los controles de seguridad mínimos y el riesgo residual aceptable, el Gerente de Proyecto elaborará un Acta de Inicio o Documento de Aprobación del Proyecto, que debe ser avalado por la Alta Dirección o la instancia designada. Este documento de aprobación debe incluir la Aceptación Formal del Riesgo Residual del Proyecto.

Ningún desarrollo, interno o tercerizado, podrá iniciar sin esta aprobación formal y sin la evidencia documentada del cumplimiento de los requisitos y controles de seguridad establecidos.

Esta aprobación deberá evidenciar el cumplimiento de los principios de gobernanza definidos en OWASP SAMM (Governance) y los controles de autorización establecidos en NIST SP 800-53.

6.2. Diseño seguro y modelamiento de amenazas

El propósito de esta fase es definir la arquitectura técnica y de seguridad del software, asegurando que se integren los controles necesarios para proteger los activos de información y cumplir los requisitos definidos en la etapa de inicio. Esta fase corresponde a las prácticas **"Design" y "Protect"** del NIST SP 800-218 (SSDF) y al dominio de Diseño Seguro de OWASP SAMM.

6.2.1. Principios de diseño seguro

Durante el diseño de la solución, los arquitectos, desarrolladores y analistas deberán aplicar los siguientes principios fundamentales:

- **Seguridad por diseño:** integrar los controles de seguridad desde la fase de conceptualización.
- **Mínimo privilegio:** los componentes, usuarios y procesos deben operar con los permisos estrictamente necesarios.
- **Defensa en profundidad:** implementar capas de control redundantes que limiten el impacto de una falla o intrusión.
- **Seguridad por defecto:** la configuración inicial del software debe ser segura sin requerir ajustes manuales posteriores.
- **Separación de funciones:** dividir las responsabilidades entre desarrollo, pruebas y operación, evitando conflictos de interés.
- **Gestión segura de dependencias:** emplear librerías, frameworks y módulos de terceros que estén actualizados, verificados y sin vulnerabilidades conocidas.
- **Validación de entradas y salidas:** asegurar que todos los datos procesados se validen y filtren conforme a las mejores prácticas de codificación segura (OWASP Top 10).
- **Privacidad desde el diseño:** minimizar el tratamiento de datos personales y proteger su confidencialidad mediante cifrado, anonimización o seudonimización.
- **Principio de reducción de superficie de ataque:** minimizar los puntos de entrada expuestos.
- **Principio de confianza cero (Zero Trust):** ningún componente o usuario debe ser considerado confiable por defecto.

Estos principios deberán ser verificados como parte de los criterios de diseño seguro definidos en ISO/IEC 27034.

6.2.2. Modelamiento de amenazas

El modelamiento de amenazas tiene como objetivo identificar, evaluar y priorizar las posibles amenazas que podrían afectar al software, considerando su arquitectura, flujo de datos y exposición.

Pasos recomendados:

1. **Definir el alcance del modelo:** delimitar los componentes, interfaces, actores y activos que intervienen en la aplicación.
2. **Elaborar diagramas de flujo de datos (DFD):** representar la interacción entre los módulos, usuarios, bases de datos, APIs y servicios externos.
3. **Identificar amenazas:** aplicar un enfoque estructurado, sugerido **STRIDE** (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege).
4. **Evaluar riesgos:** estimar la probabilidad e impacto de cada amenaza según la clasificación de criticidad del proyecto.
5. **Definir controles de mitigación:** asignar medidas técnicas o administrativas para eliminar, reducir o monitorear las amenazas detectadas.
6. **Registrar resultados:** documentar las amenazas, vulnerabilidades, controles y responsables, la cual formará parte del expediente del proyecto.

El modelamiento de amenazas deberá actualizarse iterativamente en entornos ágiles, especialmente ante cambios en el alcance, arquitectura o funcionalidades durante los Sprints. Esta actividad se alinea con las prácticas de análisis de riesgos del NIST SP 800-30 y con las prácticas de verificación de OWASP SAMM.

6.2.3. Controles de diseño y arquitectura

Durante la definición de la arquitectura del software se deberán incorporar los siguientes controles mínimos:

- **Control de acceso basado en roles (RBAC)** y políticas de autenticación robusta (multi-factor cuando aplique).
- **Cifrado de datos sensibles** en tránsito (TLS 1.2 o superior) y en reposo (AES-256 o equivalente).
- **Gestión segura de claves y contraseñas** (no incrustadas en código, uso de vaults o servicios de secretos).

- **Protección contra inyección y manipulación de datos** (validaciones en servidor, consultas parametrizadas, sanitización).
- **Protección de sesiones** (expiración automática, invalidación en cierre, tokens firmados).
- **Registro y trazabilidad** (logs auditables, inmutables y con retención definida).
- **Segregación de ambientes** (desarrollo, pruebas, producción), con controles de acceso independientes y datos anonimizados en entornos no productivos.
- **Diseño de API seguro** (control de origen, tokens válidos, limitación de tasa de peticiones, políticas CORS seguras).
- **Gestión de errores y excepciones controladas**, evitando mensajes que revelen información sensible.

La autorización de excepciones sigue un esquema escalonado según el nivel de riesgo:

- **Riesgo bajo o medio:** Puede ser aprobado por el Gerente de Proyecto, con validación del Responsable de Seguridad de la Información.
- **Riesgo alto o crítico:** Debe ser aprobado por el Comité de Seguridad de la Información o la Alta Dirección, garantizando independencia en la toma de decisiones.

Este proceso se rige por un enfoque basado en riesgo, asegurando que ninguna excepción comprometa de manera desproporcionada los activos de información. Asimismo, asegura segregación de funciones y evita que la misma persona que ejecuta el proyecto apruebe riesgos sin control. No se aprueban excepciones sin análisis de riesgo ni sin medidas de mitigación.

Los controles definidos deberán alinearse con el catálogo del NIST SP 800-53, particularmente en lo relacionado con control de acceso, protección de comunicaciones, gestión de configuraciones y auditoría. En sistemas que procesen datos personales, se deberán implementar controles de seudonimización, anonimización y limitación del tratamiento conforme al RGPD.

6.2.4. Validación del diseño

- El **Arquitecto de Soluciones** y el **Responsable de Seguridad de la Información** deberán revisar y aprobar el diseño técnico y de seguridad antes de iniciar la fase de desarrollo.
- Las revisiones deben incluir una verificación de que:
    - Todos los requisitos de seguridad y privacidad están implementados en la arquitectura.
    - Las amenazas identificadas cuentan con controles de mitigación definidos.

- Se documentaron las justificaciones para excepciones o limitaciones de control.

La validación deberá incluir evidencia documentada conforme a los requisitos del SGSI, garantizando su disponibilidad para auditorías internas y externas. En entornos ágiles, esta validación podrá realizarse mediante revisiones incrementales de arquitectura (Architecture Review) al cierre de cada Sprint relevante.

6.3. Desarrollo y codificación segura

Esta fase comprende la construcción del software conforme a los requerimientos aprobados y al diseño técnico validado. Para ello, se debe garantizar la aplicación de prácticas de codificación segura, control de versiones y verificación continua de seguridad. Esta fase se alinea con las prácticas de desarrollo seguro definidas en el NIST SP 800-218 (SSDF), particularmente en los dominios “Produce Software” y “Protect Software”, integrando controles de seguridad dentro del proceso de construcción mediante un enfoque DevSecOps. En entornos ágiles, estas actividades se ejecutan de forma iterativa dentro de los Sprints, asegurando la incorporación continua de controles de seguridad.

6.3.1. Principios generales de codificación segura

Durante el desarrollo, los programadores, analistas y revisores deberán cumplir con los siguientes principios y buenas prácticas:

- Cumplir los estándares de codificación segura definidos por la organización, basados en OWASP Top 10, NIST SSDF y ISO/IEC 27034. Se deben usar guías de codificación segura específicas para el lenguaje/framework utilizado.
- Evitar vulnerabilidades comunes como inyecciones, exposición de datos sensibles, fallas de autenticación, configuración insegura o errores de control de acceso.
- Implementar validaciones de entrada y salida para todos los datos provenientes de usuarios, sistemas externos o interfaces.
- Utilizar librerías, frameworks y dependencias confiables, verificando sus versiones y vulnerabilidades conocidas mediante herramientas automatizadas. Se sugiere el uso de OWASP Dependency-Check, herramienta que proporciona Análisis de Composición de Software (SCA) desde la línea de comandos.
- No incluir credenciales, claves o información sensible en el código fuente; usar mecanismos de gestión segura de secretos o vaults.
- Proteger la información en tránsito y en reposo mediante cifrado (TLS 1.2 o superior, AES-256, RSA-2048 o superior).
- Registrar los eventos relevantes (errores, accesos, operaciones críticas) siguiendo las políticas de auditoría y retención de logs.

- Manejar los errores y excepciones de forma controlada, sin exponer información técnica o sensible en las respuestas o interfaces.
- Asegurar la modularidad y mantenibilidad del código, permitiendo revisiones y actualizaciones seguras.
- Aplicar principios de revisión continua y mejora del código conforme a las prácticas de OWASP SAMM (Implementation y Verification).
- Asegurar que el código desarrollado cumpla con controles definidos en el NIST SP 800-53, especialmente en lo relacionado con validación de entradas, autenticación, control de acceso y auditoría.
- En entornos ágiles, las prácticas de codificación segura deben formar parte de la Definición de Hecho (Definition of Done) de cada historia de usuario.

6.3.2. Control de versiones y repositorios

- Todo el código fuente deberá almacenarse en repositorios controlados y seguros (por ejemplo, Git u otro sistema aprobado).
- Cada versión, modificación o “commit” debe estar asociada a un registro o ticket en el sistema de gestión de cambios.
- Se deben aplicar las siguientes medidas de control:
    - **Asignación de permisos** diferenciados según el rol (lectura, escritura, aprobación).
    - **Revisión por pares (peer review)** antes de fusionar cambios al entorno principal (*merge request* o *pull request*). Debe incluir una verificación mínima de los controles de seguridad aplicados.
    - **Etiquetado (tagging)** de versiones liberadas y control de ramas (*branches*) para desarrollo, pruebas y producción.
    - **Protección del repositorio** mediante autenticación multifactor y políticas de acceso mínimo.
- Los repositorios de código deben considerarse activos críticos dentro del SGSI y deben estar sujetos a controles de acceso, monitoreo y auditoría.
- Toda actividad en el repositorio debe ser trazable y auditable, conforme a los controles de registro definidos en el NIST SP 800-53 (familia AU - Audit and Accountability).
- Las ramas de desarrollo deben alinearse con la gestión de Sprints, permitiendo la integración continua de funcionalidades y controles de seguridad.

El líder técnico debe asegurar la integridad y trazabilidad del código almacenado en los repositorios.

6.3.3. Verificación de seguridad durante el desarrollo

- El equipo de desarrollo debe realizar revisiones de código seguras (code review) enfocadas en identificar fallas técnicas y de seguridad.
- Se recomienda integrar herramientas de análisis automático, tales como:
    - **SAST (Static Application Security Testing):** detección de vulnerabilidades en código fuente.
    - **DAST (Dynamic Application Security Testing):** identificación de fallas en tiempo de ejecución.
    - **IAST (Interactive Application Security Testing):** validación combinada durante las pruebas de integración.
- Los resultados de estos análisis deberán documentarse y tratarse conforme al procedimiento de Gestión de Cambios y Gestión de Vulnerabilidades.
- Ningún código con vulnerabilidades que superen el nivel de riesgo residual aceptable definido en la fase de diseño podrá avanzar a la fase de pruebas o liberación sin la debida remediación y validación.
- Estas herramientas deberán integrarse en pipelines de integración continua (CI/CD), permitiendo la detección temprana de vulnerabilidades (shift-left security).
- La ejecución de estas pruebas deberá alinearse con las metodologías definidas en la **OWASP Testing Guide**.
- Los umbrales de aceptación de vulnerabilidades deberán definirse conforme al apetito de riesgo establecido en el SGSI.

6.3.4. Controles técnicos específicos (basados en OWASP Top 10)

Los desarrolladores deberán garantizar la aplicación contenga al menos los siguientes controles de seguridad:

Riesgo OWASP Top 10 (2021)	**Medidas preventivas recomendadas**
A01 – Control de acceso roto	Aplicar controles RBAC, negar por defecto, validar permisos en servidor.
A02 – Fallos criptográficos	Usar algoritmos robustos, no almacenar contraseñas en texto claro, gestionar claves.
A03 - Inyección	Parametrizar consultas SQL/NoSQL, evitar concatenación de datos, validar entradas.
A04 - Diseño inseguro	Realizar modelamiento de amenazas y revisión arquitectónica.
A05 - Configuración de seguridad incorrecta	Deshabilitar servicios no usados, revisar cabeceras HTTP, gestionar parches.
A06 - Componentes vulnerables	Mantener inventario de dependencias y aplicar actualizaciones periódicas.
A07 - Fallas de identificación y autenticación	Implementar MFA, política de contraseña segura, control de sesiones.

A08 - Fallas en integridad de software y datos	Validar integridad con firmas o hash, controlar cargas y archivos externos.
A09 - Fallas en registro y monitoreo	Implementar logging seguro e inmutable y alerta de incidentes.
A10 - SSRF / exposiciones de servicios	Validar URLs externas, restringir llamadas de red, filtrar direcciones IP.

6.3.5. Control de calidad y evidencia

- Cada entrega parcial o *build* deberá pasar por un control de calidad y pruebas funcionales y de seguridad antes de su integración.
- Los **Analistas de área / Analista I+D** y **Analista de Seguridad de la Información** registrará los resultados de las pruebas y los defectos detectados en el repositorio correspondiente.
- La **Coordinación de Calidad y Cumplimiento (QA)** verificará que se documenten las evidencias de revisión, remediación y validación de seguridad antes de autorizar el paso a la siguiente fase.
- Toda la evidencia generada deberá gestionarse como información documentada del SGSI, garantizando su integridad, disponibilidad y trazabilidad.
- En entornos ágiles, estas validaciones se realizarán al cierre de cada Sprint como parte de la revisión de incrementos (Sprint Review).

6.3.6. Confidencialidad y propiedad intelectual

- Todo el personal involucrado en el desarrollo deberá cumplir con los acuerdos de confidencialidad y propiedad intelectual establecidos por la organización.
- Todo el código fuente, documentación, manuales, componentes, diseños, configuraciones y demás productos derivados del desarrollo pertenecen a **[NOMBRE DE LA EMPRESA]**, quien conserva la totalidad de los derechos de uso, reproducción, modificación y distribución, salvo pacto contractual expreso que disponga lo contrario.
- En caso de tratamiento de datos personales, se deberán cumplir los principios del RGPD, especialmente en lo relacionado con confidencialidad, integridad y limitación del tratamiento

6.4. Pruebas de seguridad

El objetivo de esta fase es validar que el software desarrollado cumple con los **requisitos funcionales, técnicos y de seguridad definidos**, asegurando que las vulnerabilidades sean identificadas, evaluadas y corregidas antes de su paso al ambiente productivo. Esta fase se alinea con las prácticas de verificación definidas en el NIST SP 800-218 (SSDF), las metodologías de OWASP Testing Guide y los

controles de evaluación del NIST SP 800-53, garantizando la validación continua de la seguridad del software.

6.4.1. Principios generales

Las pruebas de seguridad deben garantizar que las aplicaciones sean resistentes frente a intentos de acceso no autorizado, manipulación de datos o explotación de vulnerabilidades conocidas, conforme a los siguientes principios:

- La seguridad debe verificarse de manera continua durante todo el ciclo de desarrollo, no solo antes del despliegue.
- Las pruebas deben combinar métodos manuales y automatizados para cubrir las distintas capas de la aplicación.
- Toda vulnerabilidad identificada debe ser registrada, evaluada y tratada conforme a los procedimientos de Gestión de Cambios y Gestión de Vulnerabilidades.
- Las vulnerabilidades identificadas deberán ser gestionadas como eventos de seguridad, pudiendo escalarse a incidentes conforme a lo establecido en el NIST SP 800-61.
- Las pruebas deben integrarse dentro del ciclo de integración y entrega continua (CI/CD).
- No se autorizará el paso a producción de ningún software que presente vulnerabilidades críticas o altas sin remediar.

6.4.2. Tipos de pruebas de seguridad

Las pruebas deben planificarse según la naturaleza del software, su criticidad y exposición, incluyendo al menos los siguientes tipos:

Tipo de prueba	**Objetivo**	**Responsable principal**	**Roles de apoyo**
Pruebas estáticas de seguridad (SAST)	Analizar el código fuente o binario para detectar vulnerabilidades antes de la ejecución.	Desarrollador / Líder Técnico (Tech Lead)	Analista de Seguridad de la Información, Coordinación de Calidad y Cumplimiento
Pruebas dinámicas de seguridad (DAST)	Evaluar la aplicación en ejecución para identificar vulnerabilidades explotables en tiempo real.	Coordinación de Calidad y Cumplimiento	Analista de Seguridad de la Información, Líder Técnico
Pruebas interactivas (IAST)	Combinar análisis estático y dinámico dentro de entornos de prueba integrados.	Analista de Seguridad de la Información	Coordinación de Calidad y Cumplimiento, Desarrolladores

Análisis de composición de software (SCA)	Detectar vulnerabilidades en librerías, dependencias o componentes de terceros.	Desarrollador	Analista de Seguridad de la Información, Líder Técnico
Pruebas de penetración (Pentesting)	Simular ataques controlados para validar la eficacia de los controles implementados.	Responsable de Seguridad de la Información	Analista de Seguridad de la Información
Revisión de configuración y despliegue	Verificar que las configuraciones de servidores, bases de datos y aplicaciones sigan políticas seguras.	Líder Técnico / Arquitecto de Soluciones	Analista de Seguridad de la Información, Coordinación de Calidad y Cumplimiento

La selección de pruebas deberá alinearse con el nivel de riesgo del sistema, conforme al análisis realizado bajo NIST SP 800-30.

6.4.3. Planificación y ejecución de pruebas

- Se elaborará un Plan de Pruebas de Seguridad, que incluirá:
  - Alcance y objetivos de las pruebas.
  - Herramientas y metodologías por utilizar.
  - Criterios de aceptación y niveles de severidad de vulnerabilidades.
  - Responsables, cronograma y entregables esperados.
- Las pruebas deberán ejecutarse en un ambiente controlado y separado de producción, garantizando la protección de datos sensibles mediante su anonimización o enmascaramiento.
- Toda evidencia generada (informes, capturas, registros, resultados de herramientas) deberá conservarse como parte del expediente del proyecto.

El plan de pruebas deberá alinearse con los controles de evaluación definidos en el NIST SP 800-53 (familia CA - Security Assessment and Authorization). En entornos que involucren datos personales, las pruebas deberán garantizar la anonimización o seudonimización conforme al RGPD.

6.4.4. Tratamiento y remediación de hallazgos

- Los hallazgos se clasificarán de acuerdo con su nivel de consecuencia (Muy alto, Alto, Medio, Menor), siguiendo la metodología definida en el SGSI.
- El equipo de desarrollo será responsable de remediar las vulnerabilidades identificadas dentro del plazo establecido por su nivel de riesgo.
- Los tiempos de remediación deberán definirse conforme a acuerdos de nivel de servicio (SLA) basados en el nivel de riesgo.

- El Responsable de Seguridad de la Información verificará la corrección y documentará la validación de remediación.
- Los hallazgos críticos deberán gestionarse como incidentes de seguridad cuando representen una amenaza activa para la organización.
- Si un hallazgo no puede ser corregido, deberá documentarse su aceptación de riesgo, aprobada por la Alta Dirección o el Comité de Seguridad de la Información.

6.4.5. Criterios de aceptación para paso a producción

Un proyecto podrá avanzar a la siguiente fase únicamente si se cumple con los siguientes criterios:

- Todas las vulnerabilidades críticas y altas están corregidas y verificadas.
- Se dispone del informe final de pruebas de seguridad aprobado por el Comité de Seguridad de la Información.
- Se cuenta con la evidencia de revisión de código y análisis técnico.
- Las configuraciones del entorno productivo cumplen las políticas de seguridad de la información.
- El Responsable de Seguridad emite el aval formal de seguridad para el despliegue.

El cumplimiento de estos criterios constituye el “Application Security Acceptance” definido en la ISO/IEC 27034. En entornos ágiles, estos criterios deberán formar parte de la **Definición de Hecho (Definition of Done)** antes de considerar completado un incremento de software.

6.5. Despliegue y separación de ambientes (Dev, QA, Producción)

El objetivo de esta fase es garantizar que la liberación del software hacia los entornos de prueba y producción se realice de manera controlada, segura y trazable, con el fin de mantener la segregación de funciones y la integridad de los entornos involucrados. Esta fase se alinea con las prácticas de despliegue y mantenimiento del NIST SP 800-218 (SSDF), los controles operacionales del NIST SP 800-53 (gestión de configuraciones, control de acceso y auditoría), y los lineamientos de seguridad de aplicaciones definidos en la ISO/IEC 27034. Asimismo, en entornos ágiles, el despliegue se realiza de manera incremental mediante ciclos iterativos (Sprints), bajo un enfoque DevSecOps.

6.5.1. Principios generales

- El proceso de despliegue deberá garantizar la integridad, autenticidad y disponibilidad del software liberado.
- El proceso de despliegue debe garantizar la trazabilidad completa del cambio, conforme a los controles de auditoría del NIST SP 800-53 (familia AU).

- Cada ambiente (Desarrollo, Pruebas/QA y Producción) debe estar claramente separado a nivel lógico, físico y de permisos de acceso.
- Los cambios o despliegues deben ejecutarse únicamente por personal autorizado y diferente al desarrollador del código.
- Se debe asegurar la segregación de funciones entre desarrollo, pruebas y operación, conforme a los principios del SGSI e ISO/IEC 27034.
- Las versiones liberadas deben ser validadas, aprobadas y documentadas conforme al procedimiento de Gestión de Cambios y a los controles del SGSI.

6.5.2. Ambientes definidos

Ambiente	**Propósito**	**Características de seguridad y control**
Desarrollo (DEV)	Espacio destinado a la construcción y pruebas iniciales de código.	- Acceso restringido a desarrolladores autorizados. - Datos anonimizados o simulados. - No debe conectarse directamente con sistemas productivos. - Herramientas de compilación y control de versiones seguras.
Pruebas / Control de Calidad (QA)	Entorno de validación funcional y de seguridad del software antes del paso a producción.	- Acceso controlado a analistas QA y seguridad. - Replicación parcial del entorno productivo sin datos reales. - Ejecución de pruebas SAST, DAST, IAST, y validación de configuración segura. - Documentación de resultados y aprobación. - Validación conforme a OWASP Testing Guide.
Producción (PROD)	Ambiente donde opera el software en condiciones reales.	- Acceso limitado solo al personal autorizado de operaciones. - Requiere autorización formal de despliegue. - Monitoreo continuo, registro de eventos y alertas de seguridad. - Control de cambios mediante tickets o solicitudes aprobadas. - Implementación de controles de monitoreo continuo y respuesta a

		incidentes conforme a NIST SP 800-61.

6.5.3. Requisitos de separación y control

- Los entornos deben mantener configuraciones independientes, evitar compartir bases de datos, usuarios o credenciales entre ellos.
- Los datos reales o sensibles no podrán utilizarse en ambientes de desarrollo o pruebas, salvo que sean previamente anonimizados o enmascarados.
- Las credenciales, claves y tokens de acceso de cada ambiente deben ser diferentes y gestionadas a través de mecanismos seguros.
- Todo despliegue debe realizarse a través de pipelines de integración continua (CI/CD) controlados, auditables y con aprobación previa.
- El gerente de proyecto o Líder Técnico / Tech Lead debe garantizar que las tareas de despliegue, validación y monitoreo sean ejecutadas por personal distinto a quien desarrolló el código.
- Cualquier modificación en el ambiente productivo requerirá un ticket de cambio aprobado, evidencia de pruebas satisfactorias y validación de seguridad.
- Las configuraciones de los entornos deben gestionarse conforme a controles de gestión de configuraciones (CM) del NIST SP 800-53.
- En caso de tratamiento de datos personales, se deberá garantizar la anonimización o seudonimización en entornos no productivos conforme al RGPD.

6.5.4. Procedimiento de despliegue

1. **Preparación del paquete de despliegue:**
    - El desarrollador genera el *build* o versión firmada y verifica su integridad.
    - Se documentan los cambios, dependencias y pasos de instalación.
    - El artefacto debe estar firmado digitalmente o contar con mecanismos que garanticen su integridad.

2. **Validación previa:**
    - El Analistas de área / Analista I+D confirma que la versión superó las pruebas funcionales y de seguridad.
    - El Responsable de Seguridad valida la inexistencia de vulnerabilidades críticas.
    - Validación conforme a los criterios de aceptación definidos en ISO/IEC 27034 (Application Security Acceptance).

3. **Aprobación formal:**
    - El Gerente de Proyecto o Líder Técnico / Tech Lead emite la autorización de despliegue y programa la ejecución.

4. **Ejecución del despliegue:**
    - Personal de operaciones o un responsable designado ejecuta el despliegue siguiendo el procedimiento aprobado.
    - El despliegue debe ejecutarse mediante pipelines automatizados (CI/CD), con controles de aprobación y registro.
    - Se registran las evidencias y bitácoras de instalación.

5. **Verificación post-despliegue:**
    - Se validan los servicios, accesos, logs y monitoreo de seguridad.
    - Se documentan incidencias y acciones correctivas, si aplica.
    - Se deberá validar la correcta integración con los sistemas de monitoreo y gestión de incidentes.

6.5.5. Monitoreo y control posterior

- Tras el despliegue, el sistema debe permanecer bajo observación durante el período de estabilización definido.
- Se deberán revisar los registros de auditoría, alertas y métricas de desempeño para detectar anomalías.
- El Responsable de Seguridad de la Información evaluará los eventos registrados y coordinará acciones de mitigación o ajuste de configuración.
- El Líder Técnico / Tech Lead mantendrá actualizados los inventarios de versiones, configuraciones y componentes desplegados.
- Los eventos de seguridad identificados durante el monitoreo deberán gestionarse conforme al proceso de gestión de incidentes definido en el NIST SP 800-61.
- Se deberán definir umbrales de alerta y mecanismos de respuesta automática para eventos críticos.

6.6. Gestión de Cambios Seguros

El propósito de esta fase es garantizar que todos los cambios realizados sobre aplicaciones, componentes, configuraciones o infraestructura de software sean planificados, evaluados, aprobados y ejecutados de forma controlada, minimizando el riesgo de incidentes, errores o vulnerabilidades en los sistemas. Esta fase se alinea con los controles de gestión de cambios del NIST SP 800-53 (familia CM), las prácticas de gestión de riesgos del NIST SP 800-30, y los lineamientos de mejora continua de OWASP SAMM, garantizando que todos los cambios sean evaluados, controlados y auditables dentro del SGSI.

6.6.1. Principios generales

- Todo cambio que afecte aplicaciones, servicios o plataformas de TI debe gestionarse conforme a este procedimiento y al procedimiento de gestión de cambios destinado por la organización.
- Ningún cambio podrá ejecutarse sin la autorización previa del Gerente de Proyecto o Líder Técnico / Tech Lead y la validación de seguridad cuando aplique.
- Los cambios deben incorporar pruebas funcionales y de seguridad antes de su liberación.
- Cada solicitud debe incluir una evaluación de riesgos, el impacto sobre la disponibilidad y la información, y un plan de reversión (rollback) en caso de fallos.
- Toda evidencia de revisión, prueba, aprobación y ejecución debe ser registrada y archivada como parte del control documental del SGSI.
- Los cambios deben ser evaluados en función del nivel de riesgo definido en el SGSI, considerando los principios de confidencialidad, integridad y disponibilidad.
- En entornos ágiles, los cambios deben gestionarse como parte del backlog del producto, siendo priorizados y planificados dentro de los Sprints.

6.6.2. Tipos de cambios

Los cambios se clasifican según su impacto y criticidad:

Tipo de cambio	**Descripción**	**Aprobación requerida**
Cambio estándar	Cambios rutinarios o programados con bajo riesgo, previamente documentados (actualización menor de versión, parches planificados).	Líder Técnico (Tech Lead) y/o Coordinación de Calidad y Cumplimiento
Cambio normal	Modificaciones que requieren análisis, pruebas y aprobación formal (mejoras, nuevas funcionalidades, ajustes en configuración).	Líder Técnico + Arquitecto de Soluciones + Analista de Seguridad de la Información, con validación de Gerente de Proyectos / PMO
Cambio urgente	Modificaciones necesarias para resolver incidentes críticos o vulnerabilidades de alto riesgo que afectan la operación.	Líder Técnico + Analista de Seguridad de la Información, con notificación y validación posterior del Director de TI (CIO) y PMO

La clasificación de cambios deberá considerar el nivel de riesgo definido conforme al NIST SP 800-30.

6.6.3. Proceso de gestión de cambios seguros

1. **Solicitud del cambio**
   - El responsable técnico o usuario solicita el cambio a través del formato de solicitud de gestión del cambio.
   - La solicitud debe incluir: descripción, motivo, impacto esperado, plan de prueba, plan de reversión y responsables.

2. **Evaluación de impacto y riesgo**
   - El Gerente de Proyectos / PMO y el Responsable de Seguridad analizan el impacto sobre la disponibilidad, confidencialidad e integridad de la información.
   - Se determina si el cambio requiere pruebas de seguridad adicionales (SAST, DAST, revisión de código, etc.).
   - El análisis deberá alinearse con la metodología de evaluación de riesgos definida en el NIST SP 800-30.

3. **Aprobación**
   - El cambio debe ser aprobado antes de su ejecución, según su clasificación.
   - En caso de desarrollos internos, el cambio debe validarse contra los requisitos de seguridad definidos en el Diseño Seguro.

4. **Ejecución controlada**
   - El cambio se implementa en un ambiente de pruebas o QA antes de producción.
   - Se documentan las acciones ejecutadas, resultados obtenidos y cualquier incidencia detectada.
   - La ejecución deberá realizarse en entornos controlados y segregados, conforme a los principios definidos en la fase de despliegue.

5. **Pruebas posteriores al cambio**
   - Se realizan pruebas funcionales y de seguridad para verificar que el cambio no haya introducido errores o vulnerabilidades.
   - Las pruebas deberán alinearse con la OWASP Testing Guide.
   - Los resultados se registran y deben ser aprobados por el Gerente de Proyectos / PMO antes de cerrar el cambio.

6. **Cierre del cambio**
   - Una vez verificado el resultado satisfactorio, se actualiza la documentación técnica y de configuración.
   - Se archiva la evidencia de pruebas, aprobaciones y resultados.

6.6.4. Controles de seguridad obligatorios en los cambios

- Validación de cumplimiento de controles definidos en el NIST SP 800-53
- Validación del código o configuración antes de cada liberación.
- Ejecución de pruebas de seguridad automatizadas y revisión de vulnerabilidades.
- Documentación de resultados y remediación de hallazgos.
- Verificación del cumplimiento de políticas de control de acceso, cifrado, trazabilidad y gestión de errores.
- Evaluación de cumplimiento con OWASP Top 10 y NIST SSDF en los desarrollos internos.
- Aprobación del Responsable de Seguridad de la Información para los cambios que involucren datos personales o información sensible.
- Evaluación del impacto en la protección de datos personales (cuando aplique), conforme al RGPD.

6.6.5. Roles y responsabilidades principales

Rol	**Responsabilidad**
Solicitante / Desarrollador	Registrar el cambio, ejecutar pruebas iniciales, documentar resultados y asegurar cumplimiento de prácticas de desarrollo seguro.
Líder Técnico (Tech Lead)	Evaluar viabilidad técnica, aprobar cambios estándar, supervisar implementación y garantizar trazabilidad.
Arquitecto de Soluciones	Evaluar impacto en arquitectura, seguridad y escalabilidad; aprobar cambios estructurales o críticos.
Gerente de Proyectos / PMO	Validar impacto en alcance, cronograma y riesgos; aprobar cambios de impacto medio o alto.
Responsable / Analista de Seguridad de la Información	Evaluar riesgos de seguridad, validar controles y aprobar cambios que afecten la seguridad o datos sensibles
Director de TI (CIO)	Autorizar cambios críticos o de alto impacto organizacional.

Todos los cambios deberán ser registrados, monitoreados y evaluados como parte del proceso de mejora continua del SGSI, con el fin de permitir la identificación de tendencias, lecciones aprendidas y oportunidades de fortalecimiento del desarrollo seguro.

6.7. Gestión de Vulnerabilidades y mejoras continuas

El objetivo de esta fase es garantizar que las aplicaciones, componentes y entornos de software mantengan un nivel adecuado de seguridad a lo largo del tiempo, mediante la identificación, tratamiento, monitoreo y mejora continua de las vulnerabilidades detectadas. Esta fase se alinea con las prácticas de mantenimiento y mejora continua del NIST SP 800-218 (SSDF), la gestión de riesgos del NIST SP 800-30, los controles de monitoreo del NIST SP 800-53 y la gestión de incidentes definida en el NIST SP 800-61, integrándose con el ciclo de mejora continua del SGSI y el modelo de madurez de OWASP SAMM.

6.7.1. Principios generales

- La seguridad del software no finaliza con su puesta en producción; requiere un seguimiento continuo para detectar y mitigar vulnerabilidades emergentes.
- Las vulnerabilidades deben ser evaluadas y tratadas oportunamente de acuerdo con su criticidad y los procedimientos del SGSI.
- La gestión de vulnerabilidades y la mejora continua deben estar alineadas con los objetivos estratégicos de seguridad de la información y del área de tecnología.
- Las actividades de monitoreo y mejora deben ejecutarse bajo un enfoque de mejora continua (ciclo PHVA) del SGSI.
- La gestión de vulnerabilidades debe integrarse con el proceso de gestión de incidentes de seguridad de la información, conforme al NIST SP 800-61.
- En entornos ágiles, las vulnerabilidades deben gestionarse como elementos del backlog, priorizadas según su nivel de riesgo.

6.7.2. Identificación y análisis de vulnerabilidades

- Las vulnerabilidades podrán identificarse mediante:
    - Herramientas automatizadas de análisis (SAST, DAST, IAST, SCA, escáneres de infraestructura o aplicaciones).
    - Pruebas de penetración periódicas realizadas internamente o por proveedores especializados.
    - Las pruebas deberán alinearse con las metodologías establecidas en la OWASP Testing Guide.
    - Se deberá realizar monitoreo continuo de fuentes de inteligencia de amenazas (CVE, NVD, fabricantes) para identificar vulnerabilidades emergentes
    - Alertas o boletines de seguridad emitidos por fabricantes, comunidades o entidades reguladoras.
    - Incidentes reportados por usuarios, auditores o partes interesadas.

Cada vulnerabilidad identificada debe documentarse indicando su origen, descripción, nivel de severidad, responsable y fecha de detección.

Una vez identificada, se implementarán las acciones apropiadas en función del impacto en la conformidad del producto o servicio:

- Corregir la no conformidad mediante reprocesamiento o reparación.
- Identificar claramente los productos y servicios afectados para evitar la entrega involuntaria de salidas no conformes.
- Informar a las partes interesadas según la severidad de la no conformidad.
- Solicitar autorización bajo concesión cuando no sea posible implementar controles correctivos, dependiendo de la naturaleza de la no conformidad.

Las acciones tomadas deberán considerar el procedimiento de acciones correctivas definido por la organización. La gestión de vulnerabilidades deberá integrarse con los procesos de gestión de no conformidades y acciones correctivas del SGSI, asegurando su trazabilidad y cierre efectivo.

6.7.3. Clasificación y priorización

Las vulnerabilidades se clasificarán según su severidad e impacto potencial, aplicando criterios de evaluación de acuerdo con el esquema definido para la evaluación de riesgos y oportunidades. La clasificación deberá alinearse con la metodología de evaluación de riesgos definida en el NIST SP 800-30.

Nivel	**Descripción**	**Plazo de remediación máximo**
Crítica	Permite acceso total o pérdida de control del sistema.	24 a 48 horas
Alta	Compromete datos sensibles o funcionalidad esencial.	5 días hábiles
Media	Requiere condiciones específicas o autenticación para explotarse.	10 días hábiles
Baja	Impacto limitado, sin riesgo directo a la seguridad o disponibilidad.	30 días hábiles

Se recomienda el uso de estándares como CVSS (Common Vulnerability Scoring System) para la asignación de severidad.

6.7.4. Tratamiento y remediación

- Cada vulnerabilidad detectada deberá contar con un plan de tratamiento que defina acciones correctivas, responsables y fechas compromiso.
- El equipo de desarrollo o mantenimiento es responsable de aplicar los parches o ajustes necesarios y documentar la corrección.
- Las acciones de remediación deberán ser verificadas mediante pruebas de seguridad posteriores (re-testing).
- Las vulnerabilidades explotadas o activas deberán gestionarse como incidentes de seguridad.
- El Responsable de Seguridad de la Información verificará la efectividad de las acciones y registrará la validación de remediación.
- Si la vulnerabilidad no puede ser corregida por restricciones técnicas o funcionales, se deberá realizar un análisis de riesgo residual y una aceptación formal del riesgo aprobada por la Alta Dirección.

6.7.5. Monitoreo y verificación continua

- Se deberán realizar revisiones periódicas de los sistemas y aplicaciones para garantizar la vigencia de los controles de seguridad.
- Los sistemas en producción deben contar con mecanismos de monitoreo activo, detección de intrusiones y alertas ante eventos anómalos.
- Los resultados de los análisis y verificaciones deben documentarse y presentarse en los comités de seguridad o revisión del área de tecnología.
- La frecuencia mínima de revisión de vulnerabilidades críticas será mensual, o antes si se publica una amenaza relevante para el entorno tecnológico.

6.7.6. Mejora continua

- Los hallazgos y lecciones aprendidas derivados de auditorías, incidentes o revisiones técnicas deberán incorporarse en los planes de mejora del área de tecnología.
- El Gerente de proyectos / PMO y el Responsable de Seguridad de la Información evaluarán periódicamente la eficacia de los controles establecidos y propondrán actualizaciones a este procedimiento.
- Se fomentará la formación continua del personal técnico en desarrollo seguro, gestión de vulnerabilidades y cumplimiento normativo.

- Las métricas de desempeño (tiempo de remediación, número de hallazgos recurrentes, resultados de pruebas) se utilizarán para medir la madurez del proceso de desarrollo seguro (OWASP SAMM).
- La mejora continua deberá alinearse con el modelo de madurez OWASP SAMM, permitiendo evaluar el progreso del desarrollo seguro.
- Las lecciones aprendidas deberán incorporarse en retrospectivas de Sprint en entornos ágiles.

6.8. REQUISITOS MÍNIMOS DE SEGURIDAD PARA SOFTWARE PROPIO O DE PROVEEDORES

El propósito de este apartado es establecer los requisitos mínimos obligatorios de seguridad de la información que deberán cumplir todos los desarrollos, adquisiciones o mantenimientos de software realizados por **[NOMBRE DE LA EMPRESA]**, ya sea de manera interna o a través de proveedores externos, con el fin de garantizar la protección de los activos de información, la confidencialidad de los datos y la integridad de los sistemas. Estos requisitos se fundamentan en los controles del NIST SP 800-53, las prácticas del NIST SP 800-218 (SSDF), el estándar ISO/IEC 27034 y las buenas prácticas de OWASP, siendo la línea base obligatoria de seguridad para el software dentro del SGSI. Todo software deberá cumplir, como mínimo, con los siguientes lineamientos:

1. **Cumplimiento normativo y contractual**
    - Cumplir con las políticas internas del SGSI y las normas de protección de datos personales (RGPD y legislación nacional aplicable).
    - Cumplir con los lineamientos de desarrollo seguro establecidos en el presente procedimiento y con los marcos de referencia reconocidos:
        - NIST SP 800-218 (SSDF)
        - NIST SP 800-53
        - OWASP SAMM / OWASP Top 10
        - ISO/IEC 27034 - Seguridad en el ciclo de vida del software
    - Incluir en los contratos cláusulas específicas sobre confidencialidad, propiedad intelectual, seguridad de la información y tratamiento de datos personales.

2. **Seguridad de acceso y autenticación**
    - Implementar mecanismos de autenticación segura (preferiblemente multifactor).
    - Aplicar controles de acceso basados en roles (RBAC) con el principio de menor privilegio.

- Forzar la rotación de contraseñas y aplicar políticas de longitud y complejidad adecuadas.
- Asegurar la gestión segura de sesiones, evitando reutilización de tokens o sesiones persistentes sin control.
- Aplicar principios de Zero Trust, validando continuamente la identidad y el contexto de acceso.

3. **Protección de datos y cifrado**
   - Cifrar la información en tránsito (TLS 1.2 o superior) y en reposo (AES-256 o equivalente).
   - Asegurar una gestión adecuada de claves y certificados, evitando su almacenamiento en código fuente o archivos de configuración.
   - Implementar mecanismos de anonimización o enmascaramiento de datos personales en entornos no productivos.
   - Garantizar el cumplimiento de principios de privacidad por diseño y por defecto (Privacy by Design & by Default).

4. **Gestión segura del código y componentes**
   - Utilizar repositorios controlados y autenticados para el manejo del código fuente.
   - Prohibir el uso de librerías o dependencias con vulnerabilidades conocidas, manteniendo inventarios actualizados (SCA).
   - Aplicar revisiones de código (*code review*) y análisis estático de seguridad (SAST) antes de cada liberación.
   - Firmar digitalmente los ejecutables, instaladores o paquetes de distribución.
   - Asegurar la integridad del software mediante controles de firma y verificación de artefactos.

5. **Validación y pruebas de seguridad**
   - Realizar pruebas de seguridad integrales antes de la entrega o liberación:
     - SAST (estático)
     - DAST (dinámico)
     - IAST / SCA (interactivo y composición)
     - Pruebas de penetración (pentesting)
   - Corregir todas las vulnerabilidades críticas y altas antes del despliegue.
   - Entregar evidencia de resultados de pruebas y remediaciones validadas.
   - Las pruebas deberán ejecutarse conforme a la OWASP Testing Guide.

6. **Configuración y despliegue seguro**
   - Cumplir con las políticas de separación de ambientes (Dev, QA, Prod) y segregación de funciones.
   - Evitar configuraciones por defecto o servicios innecesarios habilitados.

- Establecer mecanismos de registro y auditoría de eventos relevantes.
- Garantizar la existencia de un plan de respaldo, restauración y contingencia aplicable al sistema.
- Aplicar controles de gestión de configuraciones (CM) y monitoreo continuo.

7. **Gestión de vulnerabilidades y actualizaciones**
   - Monitorear y aplicar actualizaciones de seguridad de manera oportuna.
   - Notificar a la organización sobre vulnerabilidades detectadas o incidentes de seguridad que puedan afectar el software.
   - Proveer soporte técnico y parches durante el ciclo de vida acordado.
   - Gestionar vulnerabilidades críticas como incidentes de seguridad cuando aplique

6.9. GESTIÓN DE PROPIEDAD INTELECTUAL Y CONFIDENCIALIDAD

El objetivo de este apartado es asegurar que los desarrollos, configuraciones, documentación, códigos fuente y demás entregables de software sean protegidos como activos de propiedad intelectual de **[NOMBRE DE LA EMPRESA]**, preservando su confidencialidad, integridad y disponibilidad, así como los derechos de uso y explotación correspondientes. Este apartado se fundamenta en los controles de protección de la información y gestión de activos del NIST SP 800-53, las prácticas de protección de artefactos del NIST SP 800-218 (SSDF), los lineamientos de seguridad de aplicaciones de la ISO/IEC 27034 y las disposiciones de protección de datos del RGPD, integrándose con el Sistema de Gestión de Seguridad de la Información (SGSI).

6.9.1. Principios generales

- Todo software, documentación técnica, manual, configuración, componente o artefacto desarrollado bajo la dirección o con recursos **de [NOMBRE DE LA EMPRESA]** constituye un activo de propiedad intelectual de la organización, salvo estipulación contractual expresa que indique lo contrario.
- La información técnica, funcional o de negocio relacionada con los proyectos de software se considera información confidencial, protegida por las políticas internas del Sistema de Gestión de Seguridad de la Información (SGSI).
- Todo el personal interno y externo que participe en actividades de desarrollo o mantenimiento debe firmar y cumplir acuerdos de confidencialidad antes de iniciar labores.
- Todos los activos de información asociados al desarrollo de software deberán ser clasificados conforme a las políticas del SGSI, definiendo niveles de acceso, almacenamiento y protección.

- El acceso a activos de propiedad intelectual deberá estar restringido bajo el principio de mínimo privilegio y necesidad de conocimiento.
- En caso de tratamiento de datos personales, se deberá garantizar el cumplimiento de los principios de confidencialidad, integridad y protección establecidos en el RGPD.

6.9.2. Protección de la propiedad intelectual

Los derechos de propiedad intelectual sobre el software desarrollado incluyen el código fuente, código compilado, documentación, diagramas, interfaces, bases de datos, configuraciones y materiales asociados.

El área de tecnología deberá asegurar que:

- Cada proyecto cuente con un registro actualizado de autoría y versiones del software.
- Los contratos con terceros incluyan cláusulas que garanticen la cesión total o parcial de derechos de uso, explotación y modificación, según corresponda.
- Se evite el uso de software, librerías o componentes sin licencias válidas o con licencias incompatibles con los fines institucionales.
- Se mantenga un inventario de licencias de software de terceros y dependencias utilizadas.
- Cualquier contribución de terceros al código esté documentada y aprobada por el Arquitecto de Software.
- El uso, reproducción o distribución del software institucional fuera de los fines de la organización requerirá autorización escrita de la Alta Dirección.
- Se garantice la integridad del código y artefactos mediante mecanismos de control de versiones, firmas digitales y validación de integridad
- Se verifique la compatibilidad de licencias de software de terceros (open source o comercial) para evitar riesgos legales o de uso indebido.
- Se deberá controlar el riesgo asociado a la cadena de suministro de software (software supply chain), validando el origen y confiabilidad de los componentes utilizados.

6.9.3. Confidencialidad y manejo de información

Toda la información generada o intercambiada durante el ciclo de desarrollo (código, credenciales, diagramas, informes, actas, resultados de pruebas, configuraciones, etc.) debe manejarse bajo los niveles de clasificación establecidos por la empresa. Se prohíbe expresamente:

- Compartir o divulgar información técnica, código o credenciales sin autorización.
- Utilizar medios personales (nubes, dispositivos externos, correos personales) para almacenar o transferir información institucional.
- Reutilizar código o componentes desarrollados internamente en proyectos ajenos sin aprobación formal.
- Los entornos de desarrollo y pruebas deben contar con controles de acceso restringido, auditoría de acciones, y protección contra fuga de información (DLP).
- Todas las acciones sobre repositorios, archivos y sistemas deberán ser registradas mediante mecanismos de auditoría.
- En caso de terminación de contratos o proyectos, los terceros deberán realizar la transferencia completa y formal de los activos de información, incluyendo código fuente, documentación y accesos, garantizando la eliminación certificada de cualquier copia no autorizada.
- El tratamiento de datos personales deberá limitarse a lo estrictamente necesario, garantizando su protección durante todo el ciclo de vida del desarrollo.

6.9.4. Cumplimiento y responsabilidades

El Gerente de Proyecto será responsable de controlar y verificar el cumplimiento de las políticas de propiedad intelectual y confidencialidad en los proyectos de desarrollo.

El Responsable de Seguridad de la Información verificará el cumplimiento de las políticas del SGSI, así como la inclusión de cláusulas contractuales relacionadas con confidencialidad, protección de datos y derechos de propiedad intelectual.

El Arquitecto de Soluciones y el Líder Técnico asegurarán que el uso de componentes, librerías y artefactos cumpla con los requisitos de licenciamiento, seguridad y control definidos.

La Alta Dirección aprobará los lineamientos de protección de activos intangibles y resolverá los conflictos relacionados con propiedad o derechos de uso.

El personal interno y contratistas deberán cumplir los acuerdos de confidencialidad, así como las sanciones internas o legales aplicables en caso de incumplimiento.

6.9.5. Documentación y evidencia

Como evidencia del cumplimiento de este apartado, se deberá conservar:

- Copias de contratos o acuerdos de confidencialidad firmados.
- Registro de autoría, versión y licencias de software desarrollado y/o entregado.
- Registros de acceso a repositorios y sistemas (logs de auditoría).
- Inventario de componentes y dependencias externas utilizadas.
- Evidencia de controles de acceso implementados sobre activos de software.
- Acta de entrega y recepción de código fuente, documentación técnica (algoritmos, interfaces, API) y claves de acceso al repositorio.
- Evidencia de entrega, transferencia y cierre de accesos al finalizar proyectos o contratos.
- Documentación de licencias y validación de cumplimiento de software de terceros.
- Usuario final - Manuales para el usuario final, los administradores del sistema y el personal de apoyo.
- Evidencia de eliminación o devolución de información al finalizar contratos.

CONTROL DE CAMBIOS		
Fecha	**Versión**	**Descripción del cambio**
DD-MMM-AAAA	00	Se crea el documento.