|  |  |  |
| --- | --- | --- |
| Sinergy & Lowells | **UNIDAD DE DESAROLLO** | |
| **UNIDAD DE DESARROLLO-PT-001** | **VERSIÓN: 01** | |
| **POLÍTICA DE DESARROLLO SEGURO** | | |
| **DOCUMENTO INTERNO PROPIEDAD DE SINERGY & LOWELLS S.A.S** | | |
| **APROBACIÓN DOCUMENTAL** | | |
| **ELABORADO POR:** | **REVISADO POR:** | **APROBADO POR:** |
| **Nombre:** Andrea Rodríguez  **Rol:** Consultora del SGSI  **Fecha:** 23/05/2024 | **Nombre:** Miguel Álzate  **Rol:** Desarrollador  **Fecha:** 28/05/2024 | **Nombre:** Oscar Amaya  **Rol:** Member Board  **Fecha:** 31/05/2024 |

Contenido

[1. OBJETIVO 3](#_heading=h.30j0zll)

[2. ALCANCE 3](#_heading=h.1fob9te)

[3. TERMINOLOGÍA 3](#_heading=h.3znysh7)

[4. NORMATIVIDAD APLICABLE 5](#_heading=h.2et92p0)

[5. DOCUMENTOS RELACIONADOS 5](#_heading=h.tyjcwt)

[6. RESPONSABILIDADES 5](#_heading=h.3dy6vkm)

[7. DISPOSICIONES 6](#_heading=h.1t3h5sf)

[7.1. CICLO DE VIDA DE DESARROLLO SEGURO 7](#_heading=h.6x5o70z1umqo)

[7.2. REQUISITOS DE SEGURIDAD DE LAS APLICACIONES 7](#_heading=h.5dbobhlih50b)

[7.3. ARQUITECTURA DE SISTEMAS SEGUROS Y PRINCIPIOS DE INGENIERÍA 9](#_heading=h.vv0jevabglz7)

[7.4. PROGRAMACIÓN SEGURA 10](#_heading=h.4mercw56p012)

[7.4.1. Entendimiento y preparación de la construcción 13](#_heading=h.7mey39bxrhem)

[7.4.2. Pruebas unitarias 13](#_heading=h.qggectnz94np)

[7.4.3. Pruebas funcionales y no funcionales 14](#_heading=h.za1ug6mw5zcl)

[7.4.4. Documentación de los componentes desarrollados 14](#_heading=h.szuigpnfkl3)

[7.4.5. Entrega de componentes desarrollados 15](#_heading=h.it268li7ukir)

[7.4.6. Revisión de los entregables 15](#_heading=h.p9p3tew7b96g)

[7.5. PRUEBAS DE SEGURIDAD EN EL DESARROLLO Y LA ACEPTACIÓN 15](#_heading=h.loxayaa5meyy)

[7.5.1. Pruebas de en ambiente de desarrollo 17](#_heading=h.xzsyx4slcagi)

[7.5.2. Medidas frente a las pruebas de seguridad de los sistemas 19](#_heading=h.h25yfjhjxwl5)

[7.5.3. Pruebas de aceptación de sistemas 20](#_heading=h.i90f82o6qoxj)

[7.5.4. Información de prueba 21](#_heading=h.m5bc4dl3sl04)

[7.6. DESARROLLO EXTERNALIZADO 22](#_heading=h.mnkkoxkcmj7u)

[7.7. SEPARACIÓN DE LOS AMBIENTES DE DESARROLLO, PRUEBAS Y PRODUCCIÓN 23](#_heading=h.pshjvqy9ue23)

[7.8. GESTIÓN DE CAMBIOS 24](#_heading=h.x875bg5usm2m)

[8. ANEXOS 25](#_heading=h.4d34og8)

[9. CONTROL DE CAMBIOS 26](#_heading=h.2s8eyo1)

# OBJETIVO

Establecer un marco integral para garantizar la seguridad en todas las etapas del ciclo de vida del desarrollo de software de la plataforma Sinergy, buscando proteger la integridad, confidencialidad y disponibilidad de la misma y de los datos procesados y almacenados, mediante la adopción de prácticas seguras, la identificación y mitigación de riesgos, y el cumplimiento de normativas y estándares de seguridad.

# ALCANCE

La presente política aplica para el ciclo de desarrollo de nuevas versiones y mejoras de la plataforma Sinergy, propiedad de Sinergy & Lowells.

# TERMINOLOGÍA

* **Amenaza:** Cualquier circunstancia o evento con el potencial de causar daño a los sistemas de información, como ataques cibernéticos, malware, errores humanos, desastres naturales, entre otros.
* **Auditoría de Seguridad:** Revisión y evaluación sistemática de los sistemas de seguridad y políticas para asegurar que cumplan con los estándares y detectar posibles vulnerabilidades.
* **Autenticación:** Proceso de verificar la identidad de un usuario, sistema o entidad, normalmente a través de credenciales como contraseñas, tarjetas inteligentes o biometría.
* **Autorización:** Proceso de determinar si un usuario o sistema tiene permiso para realizar una acción o acceder a un recurso específico.
* **Ciclo de Vida del Desarrollo de Software (SDLC):** Un marco que describe las fases involucradas en la creación de software, desde la planificación y diseño hasta la implementación, pruebas, despliegue y mantenimiento.
* **Confidencialidad:** Propiedad de la información que garantiza que los datos solo sean accesibles a las personas autorizadas y se protejan contra accesos no autorizados.
* **Control de Acceso:** Medidas y mecanismos de seguridad que regulan quién puede ver o utilizar recursos en un entorno de computación.
* **Desarrollo Seguro:** El proceso de diseñar, implementar y mantener software con medidas de seguridad integradas en cada etapa del ciclo de vida del desarrollo, para proteger contra vulnerabilidades y amenazas cibernéticas.
* **Disponibilidad:** Propiedad que garantiza que los sistemas y datos estén accesibles y utilizables por los usuarios autorizados cuando se necesiten.
* **Incidente de Seguridad:** Evento que compromete la seguridad de un sistema o datos, como una brecha de datos, un ataque de malware, o un acceso no autorizado.
* **Integridad:** Propiedad que asegura que los datos y sistemas sean exactos, completos y estén protegidos contra modificaciones no autorizadas.
* **Mitigación:** Acciones tomadas para reducir la probabilidad o impacto de un riesgo, mediante la implementación de controles y medidas de seguridad.
* **Normativa de Seguridad:** Conjunto de reglas y regulaciones que dictan cómo deben ser gestionados y protegidos los sistemas y datos para cumplir con las leyes y estándares de la industria.
* **Parche de Seguridad:** Actualización de software diseñada para corregir vulnerabilidades y mejorar la seguridad del sistema.
* **Pruebas de Penetración (Pentesting):** Evaluación de seguridad que simula ataques reales para identificar y explotar vulnerabilidades en un sistema, con el fin de mejorar la seguridad del mismo.
* **Riesgo:** La probabilidad de que una amenaza explote una vulnerabilidad, causando impacto negativo en la confidencialidad, integridad o disponibilidad de los sistemas **y datos.**
* **Vulnerabilidad:** Debilidad o falla en un sistema de software que puede ser explotada por una amenaza para obtener acceso no autorizado, causar interrupciones o comprometer la integridad de los datos.

# NORMATIVIDAD APLICABLE

* ISO/IEC 27001:2022 – Requisitos para el Sistema de Gestión de Seguridad de la Información, Ciberseguridad y Protección de la Privacidad.
* ISO/IEC 27001:2022 – Anexo A – Controles de Seguridad de la Información, Ciberseguridad y Protección de la Privacidad.
* ISO/IEC 27002:2022 Guía para la implementación de controles de Seguridad de la Información, Ciberseguridad y Protección de la Privacidad.

# DOCUMENTOS RELACIONADOS

* [SISTEMAS DE GESTIÓN-PT-002 - Políticas Específicas Seguridad de la Información.](https://drive.google.com/drive/folders/1AW3cEr0QLgivOBxqVjCHET0P49Jftuto)
* [SISTEMAS DE GESTIÓN-PD-008 - Procedimiento de Gestión de Cambios de TI](https://drive.google.com/drive/folders/1Ar3uEw5heAovlR7WzeUUO-xxWtWzYs3f)
* [UNIDAD HELPDESK-PD-001 - Gestión de pruebas](https://drive.google.com/drive/folders/17FF6u4rIjthqOdz4r3Q6KV4PTO0d0pna)

# RESPONSABILIDADES

* **Member Board:** 
  + Fomentar una cultura organizacional que priorice la seguridad de la información dentro del desarrollo de sistemas y aplicaciones, incluyendo el software que sea adquirido.
  + Aprobar y respaldar la política de desarrollo seguro.
  + Proveer capacitación y concienciación en seguridad para el personal relacionado con el desarrollo de software.
* **Equipo de desarrollo de software:** 
  + Integrar prácticas de desarrollo seguro en el ciclo de vida del desarrollo de software (SDLC).
  + Asegurar que los desarrolladores sigan las directrices de seguridad establecidas.
  + Coordinar con el equipo de seguridad para abordar vulnerabilidades identificadas.
  + Aplicar prácticas de codificación segura.
  + Realizar revisiones de código para identificar posibles vulnerabilidades.
  + Participar en sesiones de capacitación en seguridad de la información.
  + Colaborar con el equipo de seguridad para resolver problemas identificados.
* **Analista de Soporte IT:** 
  + Realizar análisis de riesgos y pruebas de penetración a la aplicación.
  + Identificar y mitigar vulnerabilidades en los sistemas y aplicaciones.
  + Monitorear y responder a incidentes de seguridad.
  + Implementar y mantener controles de seguridad en la infraestructura de TI.
  + Gestionar parches y actualizaciones de seguridad para todos los sistemas.
  + Monitorear la infraestructura para detectar y responder a incidentes de seguridad.
  + Colaborar con el equipo de desarrollo y de seguridad para asegurar el despliegue seguro de aplicaciones.
* **Responsable del SGSI:** 
  + Desarrollar, implementar y mantener la política de desarrollo seguro.
  + Supervisar las evaluaciones de riesgos y las auditorías de seguridad.
  + Asegurar el cumplimiento de normativas y estándares de seguridad.
  + Reportar a la alta dirección sobre el estado de la seguridad y las necesidades de mejora.
  + Realizar auditorías periódicas para evaluar el cumplimiento de la política de desarrollo seguro.
  + Identificar áreas de mejora y recomendar acciones correctivas.
  + Informar a la alta dirección sobre los hallazgos y el estado de cumplimiento.

# DISPOSICIONES

**Lineamientos generales**

* Se debe asegurar que las prácticas de seguridad sean una parte fundamental de cada fase del desarrollo, desde la planificación hasta el mantenimiento.
* Se debe proveer formación continua para mantener a los desarrolladores y al personal de TI actualizados sobre las últimas amenazas y técnicas de desarrollo seguro.
* Se deben implementar evaluaciones periódicas de riesgos y establecer procedimientos para identificar y mitigar vulnerabilidades y amenazas.
* Se debe garantizar que todos los desarrollos cumplan con los estándares de seguridad y las normativas legales aplicables.
* Se deben establecer mecanismos de monitoreo y auditoría continua para evaluar y mejorar las medidas de seguridad implementadas.

## CICLO DE VIDA DE DESARROLLO SEGURO

* Se debe velar porque el desarrollo interno o externo de los sistemas de información cumpla con los requerimientos de seguridad esperados, con las buenas prácticas para desarrollo seguro de aplicativos, así como con metodologías para la realización de pruebas de aceptación y seguridad al software desarrollado. Además, se asegurará que todo software desarrollado o adquirido, interna o externamente cuenta con el nivel de soporte requerido por la compañía.
* Se deberán tener en cuenta los siguientes aspectos:
  + Tener separados los ambientes de desarrollo, prueba y producción.
  + Contar con una metodología de desarrollo de software en la que se indiquen las pautas de codificación segura para cada lenguaje de programación utilizado.
  + Identificar los requisitos de seguridad en las fases de levantamiento de requerimientos y diseño.
  + Realizar identificación y seguimiento de puntos de seguridad de la información en la gestión de proyectos.
  + Contemplar realizar pruebas en las aplicaciones, tales como escaneo del código y pruebas de penetración.
  + Contar con repositorios seguros para el código fuente y la configuración de elementos de red, software, entre otros.
  + Proteger el acceso al código fuente.
  + Controlar las versiones del software.
  + Garantizar la competencia de los desarrolladores en seguridad de las aplicaciones, así como para prevenir, encontrar y reparar vulnerabilidades.

## REQUISITOS DE SEGURIDAD DE LAS APLICACIONES

* Sinergy & Lowells asegurará que el software adquirido y/o desarrollado al interior de la compañía, como por terceras partes, cumplirá con los requisitos de seguridad y calidad establecidos por él.
* Se deberán incluir requisitos de seguridad en la definición de requerimientos y, posteriormente se asegurará que estos se encuentren generados a cabalidad durante las pruebas realizadas sobre los desarrollos del software construido.
* Los requisitos de seguridad de la información, cuando corresponda, deberán incluir:
  + Requisitos de seguridad para la validación de identidades.
  + Segregación de permisos de acceso y de privilegios sobre los datos y funcionalidades de las aplicaciones.
  + Resiliencia contra ataques maliciosos o denegación de servicios.
  + Requisitos legales, reglamentarios y contractuales aplicables y suscritos por Sinergy & Lowells para la generación, almacenamiento, procesamiento y/o transferencia de la información.
  + Protección de datos de las partes interesadas.
  + Requisitos para la protección de información confidencial.
  + Protección de datos en proceso, en tránsito y/o en reposo.
  + Necesidades de cifrado en la comunicación o transmisión de la información entre las partes interesadas.
  + Controles automatizados, como, por ejemplo: límites de aprobación o doble aprobación.
  + Restricciones relacionadas con la introducción de texto libre.
  + Requisitos derivados de los procesos de operaciones, tales como registro y requisitos de no repudio (en caso de autorizaciones de tratamiento de datos personales, entre otros).
  + Requisitos de interfaces para registro o sistemas de detección de fuga de datos.
  + Manejo de mensajes de error.
  + Mecanismos para la identificación de falta de integridad.
  + Métodos criptográficos para la autenticación y seguridad de la transferencia de datos.
* Para determinar qué controles deberá tener las aplicaciones desarrolladas y/o software adquirido, se debería realizar un análisis de riesgos para determinar con mayor exactitud y ser más acertados en la definición de los controles a implementar que corresponda, según sea el caso.
* En el caso de aplicaciones de servicios transaccionales, se debe asegurar que se hace un debido control en la gestión de identidades, integridad de la información procesada y/o intercambiada, así como métodos de verificación de la integridad, procesos de autorización para aprobar y firmar registros de transacciones, SLA's y otros requisitos contractuales.
* Para las aplicaciones de pagos electrónicos, se debe asegurar la confidencialidad e integridad de la información, los mecanismos de verificación y validación de los pagos realizados por los clientes, controles para resguardar la información del pago, así como para evitar pagos duplicados, entre otros.

## ARQUITECTURA DE SISTEMAS SEGUROS Y PRINCIPIOS DE INGENIERÍA

* Sinergy & Lowells garantizará que la plataforma se diseña, implementa y opera de forma segura a lo largo de todo su ciclo de desarrollo, teniendo en cuenta las diferentes capas de arquitectura (negocios, datos, aplicaciones y tecnología).
* Las nuevas tecnologías deben analizarse para detectar riesgos de seguridad de la información y el diseño debe realizarse frente a patrones de ataque conocidos.
* La arquitectura debe contemplar los siguientes aspectos:
  + Técnicas de autenticación de usuarios, controles de sesión seguro, validación y saneamiento de datos.
  + Controles para proteger la seguridad de la información y de sistemas contra amenazas conocidas.
  + Controles para prevenir, detectar o responder a eventos de seguridad.
  + Cuando aplique, encriptación de la información confidencial, verificación de la integridad y firma digital.
  + Controles de seguridad en la infraestructura que soporta las aplicaciones, como, por ejemplo: gestión de acceso e identidades, prevención de fuga de datos, entre otros.
  + Endurecimiento de los sistemas.
  + Realizar revisión del diseño para identificar vulnerabilidades de la seguridad de la información, asegurar que se especifiquen y cumplan los controles de seguridad de la información.
  + Revisar mínimo una (1) vez al año o cuando se produzcan cambios, la arquitectura de la solución para asegurar que permanezca actualizada para combatir nuevas amenazas potenciales y seguir siendo aplicable conforme hay avances en la tecnología.
* Aplicar la arquitectura de seguridad al desarrollo subcontratado de aplicaciones mediante contratos o acuerdos contractuales entre Sinergy & Lowells y los proveedores de estos servicios. Estos contratos deben especificar claramente las políticas y estándares de seguridad que deben seguirse, incluyendo la implementación de prácticas de desarrollo seguro, la realización de evaluaciones de riesgos, y la obligación de mitigar cualquier vulnerabilidad identificada.
* Incluir cláusulas que aseguren el cumplimiento de normativas legales aplicables y permitan auditorías periódicas para verificar el cumplimiento de estos requisitos.

## PROGRAMACIÓN SEGURA

* Sinergy & Lowells establecerá una metodología y procedimientos detallados con los lineamientos para el desarrollo seguro, los cuales deben abarcar tanto los componentes de software de terceros como el software de código abierto.
* Se debe incluir dentro de dicha metodología directrices específicas para la selección, evaluación, integración y monitoreo de estos componentes, asegurando que cumplan con los estándares de seguridad y normativas legales aplicables. Además, debe contemplar la realización de auditorías periódicas y la aplicación de parches y actualizaciones de seguridad necesarias para mantener la integridad y seguridad del software.
* Sinergy & Lowells deberá monitorear continuamente las amenazas presentes en el mercado y en el ciberespacio, actualizando regularmente la información sobre vulnerabilidades de software. Esto permitirá establecer principios de programación segura que estén orientados a combatir el panorama de amenazas en constante evolución. Estos principios deben integrarse en el ciclo de vida del desarrollo de software y ser actualizados periódicamente para asegurar que las aplicaciones desarrolladas sean resilientes frente a las nuevas amenazas y vulnerabilidades emergentes.
* Los principios de programación segura deben emplearse tanto en el desarrollo de nuevas aplicaciones como en los cambios de versiones de las ya existentes. Esto se aplica tanto a las aplicaciones desarrolladas internamente como a aquellas subcontratadas externamente. Estos principios deben integrarse en cada etapa del ciclo de vida del desarrollo de software para garantizar que todas las aplicaciones mantengan un alto nivel de seguridad y sean capaces de resistir las amenazas y vulnerabilidades emergentes.
* Se deben tener en cuenta prácticas y defectos de codificación comunes e históricos que son conocidos por conducir a vulnerabilidades de seguridad de la información. Es fundamental aprender de estos errores pasados para evitar su repetición, incorporando este conocimiento en las directrices de desarrollo seguro y en la formación continua de los desarrolladores. Esto ayudará a fortalecer la resiliencia del software contra posibles amenazas y a mejorar la calidad general de las aplicaciones desarrolladas.
* Sinergy & Lowells deberá mantener y actualizar constantemente las herramientas de desarrollo y resguardo del código. Esto garantiza que las herramientas utilizadas para desarrollar, revisar y proteger el código estén siempre alineadas con los últimos estándares de seguridad y tecnologías disponibles.
* Sinergy & Lowells deberá asegurar la competencia del personal de programación, proporcionando formación y desarrollo profesional continuo para garantizar que los desarrolladores estén al tanto de las mejores prácticas de seguridad y las últimas técnicas de programación segura. Además, se debe fomentar un ambiente donde el intercambio de conocimientos y la colaboración entre el personal promuevan un nivel óptimo de competencia en el desarrollo de software seguro.
* Se debe incluir el modelado de amenazas dentro del diseño y la arquitectura de las aplicaciones, identificando y analizando posibles amenazas y vulnerabilidades desde las etapas iniciales del desarrollo, permitiendo así diseñar contramedidas efectivas para mitigar los riesgos de seguridad.
* Integrar el modelado de amenazas en el proceso de diseño y arquitectura asegura que las aplicaciones estén preparadas para hacer frente a las amenazas potenciales desde su concepción hasta su implementación y más allá.
* Durante el proceso de programación, es fundamental implementar técnicas de revisión por pares, pruebas de requisitos de seguridad y otras buenas prácticas. Estas actividades deben garantizar que el código desarrollado cumpla con los estándares de seguridad establecidos y se encuentre libre de vulnerabilidades. La revisión por pares permite detectar posibles errores y debilidades en el código, mientras que las pruebas de requisitos de seguridad verifican que las funcionalidades implementadas cumplan con los criterios de seguridad definidos.
* Se deberán realizar pruebas tanto durante como después del desarrollo de software. Esto incluye pruebas de seguridad estáticas para identificar vulnerabilidades de seguridad en el código. Estas pruebas se deben llevar a cabo mediante herramientas especializadas que analicen el código fuente en busca de posibles vulnerabilidades y debilidades de seguridad.
* Se deberán realizar un análisis de los errores de programación comunes y verificar que hayan sido mitigados, identificando patrones de errores históricos y conocidos que puedan conducir a vulnerabilidades de seguridad, y luego asegurarse de que se hayan aplicado medidas correctivas o mitigaciones adecuadas en el código.
* Las actualizaciones de software deben ser empaquetadas e implementadas de forma segura, garantizando la integridad y autenticidad de los paquetes de actualización, así como protegiendo los procesos de distribución y despliegue contra posibles ataques o manipulaciones maliciosas.
* Se deben gestionar las vulnerabilidades de seguridad de la información informadas. Los errores y los ataques sospechosos deben registrarse y revisarse periódicamente para realizar los ajustes necesarios al código.
* El código fuente debe protegerse contra accesos no autorizados mediante el uso de herramientas de gestión de la configuración y control de versiones.
* En caso de emplear bibliotecas y herramientas externas, se debe asegurar que las mismas cuenten con licencias, seguridad e historial de dichos componentes externos, garantizar que dicho software proviene de fuentes confiables, asegurar que haya disponibilidad a largo plazo de los recursos y artefactos de desarrollo, entre otros.
* Cuando sea necesario modificar un paquete de software, se debe tener en cuenta los riesgos que implicaría que los controles incorporados y los procesos de integridad se vean comprometidos, si es necesario obtener el consentimiento del fabricante, la posibilidad de obtener cambios en los sistemas por parte del proveedor como actualizaciones del software, medir el impacto y definir los responsables por el mantenimiento futuro del software, compatibilidad con otros componentes, gestión de cambios, entre otros.
* Se debe garantizar que el código sea resistente a la manipulación y que las bases de datos estén protegidas en entornos seguros.

### Entendimiento y preparación de la construcción

* Una vez aprobado el requerimiento y el planteamiento de la solución del mismo, el líder técnico es el responsable de contextualizar al equipo de desarrollo.
* Este proceso incluye la asignación de tareas específicas al equipo y la estructuración detallada del plan de trabajo.
* La personalización del plan de trabajo se realiza de acuerdo al tipo de solicitud, asegurando así una preparación sólida y alineada con los objetivos del requerimiento.

|  |  |
| --- | --- |
| **Tipo de Requerimiento** | **Responsable** |
| **Incidente** | Analista de Mesa de Ayuda |
| **Proyectos** | Líder de Consultoría |
| **Mejora Evolutiva** | Desarrollador |

* Durante la construcción del código fuente, el equipo de desarrollo se alineará rigurosamente a las buenas prácticas y lineamientos de programación y seguridad, establecidas en la Arquitectura de Referencia de TI. Esto garantizará la creación de un código fuente legible y comprensible, facilitando así futuras modificaciones.
* Se deberá realizar un control del código fuente para gestionar de manera efectiva el código fuente y la generación de versiones, asegurando así la trazabilidad y calidad del desarrollo de la solución.

### Pruebas unitarias

* Las pruebas unitarias representan un mecanismo esencial para verificar el comportamiento esperado de un componente de software en diversos escenarios. Este proceso, debe ser llevado a cabo por el equipo de desarrollo, en el cual asegura la cobertura de la funcionalidad requerida, permitiendo así la identificación y resolución proactiva de errores antes de avanzar a las pruebas funcionales.
* La salida de estas pruebas será entregada al equipo de pruebas correspondiente, ya sea un proveedor externo o interno, según sea el requerimiento.
* Esta colaboración debe asegurar la transición fluida entre las fases de desarrollo y prueba, fortaleciendo la calidad del producto final al anticipar y abordar eficientemente cualquier inconveniente identificado durante las pruebas unitarias.

### 

### Pruebas funcionales y no funcionales

* El equipo de pruebas (Mesa de Ayuda), son los encargados de llevar a cabo pruebas exhaustivas sobre los componentes del software en su totalidad, evaluando también su interacción con el entorno de ejecución del sistema.
* En esta fase, se deben ejecutar los casos de prueba definidos durante la etapa de diseño en el ciclo de vida del desarrollo de software, con el objetivo de identificar y abordar posibles errores antes de la entrega a calidad.
* Es esencial que, al finalizar esta etapa, exista evidencia concreta de la ejecución de las pruebas, la cual deberá ser presentada como parte integral del entregable. Este enfoque garantizará un proceso de prueba completo y documentado, contribuyendo a la calidad y confiabilidad del software antes de su implementación final.

### Documentación de los componentes desarrollados

* Es responsabilidad del equipo de desarrollo llevar a cabo la creación y entrega de la documentación correspondiente a los componentes desarrollados.
* Este proceso debe asegurar una documentación integral y coherente que respalda la comprensión y mantenimiento efectivo de los componentes, facilitando la continuidad del desarrollo y la colaboración entre las áreas.

### 

### Entrega de componentes desarrollados

* El equipo de desarrollo debe llevar a cabo la consolidación de todos los entregables al líder técnico, como parte del avance a la siguiente fase de la solución. Estos entregables incluyen:
  + Componentes desarrollados y cargados en la herramienta de control de fuentes y versiones.
  + Documentación técnica detallada.
  + Manual de instalación de los componentes desarrollado.
  + Evidencia documentada de la ejecución de las pruebas.
* Esta compilación de elementos garantiza una transición eficiente y completa hacia la siguiente fase, proporcionando al líder técnico los recursos necesarios para una evaluación integral y un progreso continuo del proyecto

### Revisión de los entregables

* El analista de servicios de aplicaciones debe emitir su visto bueno sobre los entregables recibidos después de verificar su completitud.
* Con esta aprobación, se procede a avanzar a la siguiente fase del ciclo de vida de software, asegurando así que todos los elementos críticos hayan sido adecuadamente revisados y validados antes de avanzar en el proceso.

## PRUEBAS DE SEGURIDAD EN EL DESARROLLO Y LA ACEPTACIÓN

* Dentro del proceso de pruebas realizado, Sinergy & Lowells debe incluir pruebas que aseguren que las mejoras a la plataforma, actualizaciones y nuevas versiones cumplen con los requisitos de seguridad.
* Para lo anterior, debe incluir, entre otras, las siguientes pruebas:
  + Pruebas de autenticación de usuarios.
  + Pruebas de restricción de accesos.
  + Pruebas de configuraciones seguras, incluido sistemas operativos, firewall, WAF, entre otros.
* Es fundamental que los sistemas de información nuevos, así como las actualizaciones y nuevas versiones, sean sometidos a pruebas exhaustivas y verificaciones tanto durante como después de los procesos de desarrollo. Estas pruebas y verificaciones deberán asegurar que los sistemas cumplan con los requisitos funcionales y de seguridad establecidos, así como con los estándares de calidad definidos.
* Las pruebas de seguridad deben ser una parte integral de las pruebas realizadas a los sistemas o componentes de los mismos.
* Se deben establecer el conjunto de requisitos de seguridad y que son necesarios probar, los cuales pueden ser funcionales o no funcionales y deben incluir, entre otros, los siguientes:
  + Funciones de seguridad: Autenticación de usuarios, restricción de accesos, uso de criptografía.
  + Programación segura.
  + Configuraciones de seguridad, incluidas firewall., WAF y otros componentes de seguridad.
* Se deben establecer planes y criterios de prueba, los cuales se determinarán con base en la importancia del software, el impacto potencial de cambios que se estén introduciendo y deben incluir:
  + Cronograma detallado de las pruebas y actividades.
  + Insumos y resultados esperados.
  + Criterios para evaluar los resultados.
* Para los desarrollos internos, las pruebas se deberán realizar inicialmente por el equipo de desarrollo y posteriormente se deben realizar pruebas de aceptación independientes para garantizar la funcionalidad esperada del software. Se deben realizar actividades de revisión de código para probar fallas de seguridad, realizar un escaneo de vulnerabilidades para identificar configuraciones inseguras y vulnerabilidades del sistema y en lo posible realizar pruebas de penetración para identificar código y diseño inseguro.
* Para el software desarrollado por terceros o comprado, se debe establecer un proceso de adquisición que incluya contratos con el proveedor en los cuales se deben abordar los requisitos de seguridad establecidos por Sinergy & Lowells. Los productos y servicios deben evaluarse teniendo en cuenta dichos criterios antes de la adquisición del software.
* Sinergy & Lowells S.A. debe exigir a los proveedores en los desarrollos de software, que deben contar con ambientes independientes tales como: pruebas, certificación y producción, estos deben contemplar controles físicos y de acceso lógico para asegurar la separación de estos. (Aplica para sistemas de información nuevos o existentes)
* Las pruebas se deben realizar en entornos de prueba separados de los entornos de desarrollo y producción, garantizando la integridad y la estabilidad de los sistemas de producción al minimizar el riesgo de impacto adverso causado por pruebas o experimentación.
* Se debe asegurar la eficacia de las pruebas mediante la evaluación y monitoreo de los ambientes de prueba, las herramientas y tecnologías empleadas las cuales deben ser similares o idénticas a la configuración de los ambientes de producción.

### Pruebas de en ambiente de desarrollo

* Los datos almacenados en el ambiente de producción no deben ser utilizados para las actividades de certificación o pruebas.
* Los datos utilizados en los ambientes de desarrollo, pruebas, certificación no deben ser los utilizados en el ambiente de producción.
* Habilitar solo los módulos, servicios, protocolos y aplicaciones que sean necesarias para el buen funcionamiento del sistema de información. Aquellos que no se utilicen deberán ser deshabilitados.
* El desarrollador es responsable de documentar cuáles son estrictamente necesarios para el correcto funcionamiento del aplicativo.
* Verificar que los aplicativos cuenten con las últimas versiones estables, tanto a nivel de software como de sistema operativo, parches de seguridad, servidor de aplicaciones, base de datos, máquina virtual de java, etc., antes del despliegue.
* Verificar que no se pueda listar los directorios de la aplicación.
* Verificar que en el servidor no se encuentren instalados módulos, extensiones, programas por defecto y que no serán usados por la aplicación.
* Controlar para que los usuarios no tengan acceso a aquellos archivos de configuración o a directorios sensibles que no puedan ser eliminados. Solo usuarios con privilegios o autorizados deberán tener acceso.
* Controlar para que todos los usuarios de la aplicación y el software que se ejecuten en el servidor (base de datos, sftp, apache, iis, jboss, tomcat, etc.) tengan los mínimos privilegios sobre el sistema.
* Garantizar que no se permita la transferencia de archivos con configuraciones del sistema a los usuarios.
* Controlar si la aplicación requiere que el usuario adjunte archivos, verificar que solo este permitido el envío de documentos con extensiones específicas, por ejemplo: doc, docx, pdf. No permitir que el usuario adjunte archivos con extensiones asp, txt, php, jsp, exe, etc.
* Validar los archivos enviados al servidor por el usuario en la estructura de su cabecera, ya que la extensión puede ser falsificada por este.
* Controlar para que los archivos que son enviados por el usuario no sean almacenados en el mismo entorno de trabajo de la aplicación, se recomienda guardarlos en un dispositivo aislado.
* Controlar y de ser posible solo permitir el cargue de archivos .pdf a los sistemas de información y, en la medida de lo posible, no deben incluir scripts.
* Controlar para que los archivos transferidos al servidor por el usuario no se almacenen con permisos de ejecución, sólo de lectura.
* No utilizar rutas específicas en los parámetros o variables, se recomienda, utilizar índices que internamente se asocien a directorios o rutas predefinidas o utilizar protocolos seguros, tales como SSH, SFTP, FTPS, VPN SSL, IP SEC, etc. Para la comunicación y transferencia de archivos.
* Todos los sistemas que implementen Web Services dentro de su funcionamiento, deben contar con mecanismos de seguridad adecuados.
* Se deben tener en cuenta los siguientes aspectos, sin limitarse a:
  + Configurar solo métodos HTTP seguros, tales como POST. No permitir el uso de los métodos DELETE, PUT, GET, y TRACE. o Implementación de soporte HTTPS para mensajes tipo SOAP.
  + Cifrado y firma de contenido XML.  Implementación de especificaciones de seguridad como: WSSecurity, WSTrust, WS-Signature, XML Encryption, SAML, ebXML, WS-policy.  Validación de entradas.  Validación de XML según el esquema definido.  Generación de logs de las transacciones de los Web Services.
  + Modificar las cabeceras HTTP para que no muestre información sobre las versiones y las aplicaciones que se encuentran en el servidor. Por ejemplo: modificar la cabecera “Server”.

### Medidas frente a las pruebas de seguridad de los sistemas

* Se deben tener en cuenta las siguientes consideraciones al realizar pruebas de seguridad:
  + Las revisiones de código pueden ser realizadas por terceros contratados para tal efecto, o por personal interno capacitado para hacerlo con el fin de asegurar la idoneidad de quien realiza dicha actividad.
  + Puede realizarse a través de herramientas automáticas o mediante técnicas manuales.
  + Los cambios en el código fuente deben ser revisados por personas diferentes al autor de este, con conocimiento en técnicas de verificación de código y desarrollo seguro.
  + Las revisiones de código deben estar orientadas a verificar que la codificación cumpla con todos los requerimientos expuestos en la presente política.
  + Las observaciones generadas en la revisión deben ser atendidos antes de la puesta en producción del sistema.
  + Los procesos de revisión deben alinearse a los requerimientos de normativas y regulaciones aplicables al Sinergy & Lowells S.A.
  + Una evaluación del riesgo de vulnerabilidades en el código debe realizarse y valorarse con los interesados en el software creado.
  + Verificar que los aplicativos cuenten con las últimas versiones estables, tanto a nivel de software como de sistema operativo, parches de seguridad, servidor de aplicaciones, base de datos, máquina virtual de java, etc., antes del despliegue.
  + Realizar por lo menos una revisión del código antes de liberarlo en producción, y poder llevar a cabo las actividades de remediación y seguimiento a estas vulnerabilidades antes que salga a producción. En este proceso se deben tener en cuenta las vulnerabilidades más importantes en la industria (OWASP Top 10, CWE Top 25, etc.).

### Pruebas de aceptación de sistemas

* Establecer programas de prueba para aceptación, escaneo de vulnerabilidades y criterios de aceptación relacionados para los sistemas de información nuevos, actualizaciones y nuevas versiones.
* Definir y ejecutar las pruebas de aceptación de software a partir de los siguientes elementos:
  + Requerimientos del usuario.
  + Requerimientos de sistema.
  + Casos de uso.
  + Procesos de negocio.
  + Atributos de calidad.
  + Reportes de análisis de riesgo.
  + Requisitos de seguridad de la información
* Incluir en las pruebas de requisitos de seguridad de la información los siguientes puntos:
  + Llevarse a cabo sobre componentes y sistemas integrados.
  + Hacer uso de herramientas automatizadas, tales como herramientas de análisis de códigos o escáneres de vulnerabilidad, y verificar que se han corregido los defectos relacionados con la seguridad.
  + Realizar una prueba de intrusión, en ambiente productivo definitivo, pero con acceso controlado con el fin de conocer y evaluar las condiciones de seguridad que rodean el ambiente final con el fin de hacer ajustes finales, el responsable de esta revisión debe ser una persona diferente al desarrollador (Por ejemplo: revisión por pares).
  + Realizar pruebas de penetración y/o análisis de vulnerabilidades antes de salir a producción.
  + Al identificar una vulnerabilidad técnica en el proceso de desarrollo de un sistema, se debe identificar el riesgo asociado y el plan de acción para remediarla.
* Las pruebas de seguridad funcional se deben basar en los requerimientos definidos con respecto a:
  + Autenticación solicitada, complejidad de contraseñas basada en la política definida en este documento y restricciones de acceso según lo diseñado en roles y permisos.
  + Bloqueo automático de cuentas de acceso.
  + Funcionalidad de captchas (verificar que el usuario que está accediendo a determinados datos es un humano y no una máquina).
  + Lo registrado en los logs, así como su almacenamiento.
  + Los mensajes de error que se deben presentar en las acciones validadas.
  + Garantizar el cumplimiento de las obligaciones legales, regulatorias contractuales establecidas.
  + Asignar responsabilidades frente a seguridad de la información, las cuales son definidas, compartidas y publicadas.

### Información de prueba

* Sinergy & Lowells deberá proteger los datos de prueba que se entregarán a los desarrolladores, asegurando que no revelen información confidencial de los ambientes de producción.
* Se debe aplicar los mismos controles de acceso lógico en los ambientes de prueba que se aplican en el ambiente de producción. Esto garantiza la consistencia en la gestión de accesos y reduce el riesgo de brechas de seguridad en los entornos de prueba que podrían afectar la integridad y confidencialidad de los datos.
* Es imprescindible contar con una autorización explícita siempre que se deba copiar información del ambiente de producción al ambiente de pruebas. Esta autorización garantiza que cualquier transferencia de datos se realice de manera controlada y conforme a las políticas de seguridad establecidas.
* Sinergy & Lowells debe mantener registros (logs) de la copia y del uso de información de producción con el fin de establecer un registro de auditoría completo. Estos registros proporcionan trazabilidad y transparencia en cuanto a quién accede y cómo se utiliza la información sensible del ambiente de producción en los entornos de prueba.
* Se deben implementar técnicas de enmascaramiento de datos para proteger la información confidencial y/o los datos personales, asegurando que los datos sensibles sean ocultados o modificados de manera que se mantenga su utilidad para propósitos de desarrollo o pruebas, pero sin revelar la información confidencial real.

## DESARROLLO EXTERNALIZADO

* En caso de contratar desarrollo por partes externas a la organización, Sinergy & Lowells deberá dirigir, monitorear y revisar las actividades relacionadas con el desarrollo para que este sea realizado conforme con las políticas, controles y requerimientos establecidos por la compañía.
* Sinergy & Lowells debe asegurarse de contar con los acuerdos de licenciamiento, la propiedad del código, los derechos de propiedad intelectual, los términos y condiciones de uso, entre otros aspectos legales y contractuales relevantes, garantizando que el desarrollo y la implementación del software se realicen conforme a las normativas y regulaciones aplicables, así como a los términos acordados con los proveedores y terceros involucrados.
* Dentro de los acuerdos se deben especificar los requisitos y acuerdos para la realización de pruebas de aceptación de calidad y entregables, así como el aseguramiento que las pruebas realizadas han contemplado los requisitos de seguridad y privacidad de los sistemas y la garantía de protección contra vulnerabilidades conocidas.
* Dentro de los acuerdos con proveedores de software o de desarrollo de software deben existir cláusulas que permitan auditar al proveedor procesos y controles de desarrollo.
* Dentro de los acuerdos con los proveedores deben quedar cláusulas relacionadas con la protección de datos personales durante el desarrollo y pruebas realizadas al software.
* Supervisar y hacer seguimiento de la actividad de desarrollo de sistemas contratados externamente.
* Verificar y hacer cumplir por parte del tercero los siguientes puntos en toda la cadena:
  + Los requisitos contractuales para prácticas seguras de diseño, codificación y pruebas.
  + El suministro del modelo de amenaza aprobado al desarrollador externo.
  + Las pruebas de aceptación para determinar la calidad y exactitud de los entregables.
  + La entrega de evidencias de que se usaron umbrales de seguridad para establecer niveles mínimos aceptables de calidad de la seguridad y de la privacidad.
  + La entrega de evidencias de que se han hecho pruebas suficientes para garantizar que el sistema está protegido contra contenido malicioso intencional y no intencional en el momento de la entrega.
  + La entrega de evidencias de que se han hecho pruebas suficientes para garantizar que el sistema está protegido contra la presencia de vulnerabilidades conocidas.
  + Derecho contractual con relación a procesos y controles de desarrollo de auditorías.
  + Documentación eficaz del ambiente de construcción usado para crear los entregables.
  + Aceptar las políticas y procedimientos que se encuentran dentro de este documento.
  + Cumplir con el procedimiento del protocolo de paso a producción
  + Realizar para todos los sistemas desarrollados, un proceso de verificación de código antes de su salida a producción, incluyendo aquellas librerías de terceros que son incluidas en el desarrollo (DLL).
* Establecer programas de prueba para aceptación y criterios de aceptación relacionados para los sistemas de información nuevos, actualizaciones y nuevas versiones.

## SEPARACIÓN DE LOS AMBIENTES DE DESARROLLO, PRUEBAS Y PRODUCCIÓN

* Sinergy & Lowells debe proveer los recursos necesarios para la implantación de controles que permitan la separación de ambientes de desarrollo, pruebas y producción, teniendo en cuenta consideraciones como: controles para el intercambio de información entre los ambientes de desarrollo y producción, la inexistencia de compiladores, editores o fuentes en los ambientes de producción y un acceso diferente para cada uno de los ambientes.
* Los sistemas de información deben tener por lo menos tres ambientes: Ambiente de Desarrollo, Ambiente de Pruebas y Ambiente Productivo.
* Es fundamental separar adecuadamente los ambientes de desarrollo y producción, ya sea en entornos físicos o virtuales distintos, asegurando la integridad y la estabilidad de los sistemas en producción al evitar interferencias o riesgos potenciales causados por pruebas o desarrollos en curso.
* Sinergy & Lowells debe definir procedimientos documentados que establezcan las reglas y autorizaciones necesarias para el despliegue de software en producción, de tal forma que se garantice que el despliegue de software se realice de manera controlada y segura, siguiendo un proceso formal que incluya la revisión y aprobación por parte de los responsables designados.
* Se debe evitar el uso de información confidencial o datos personales no enmascarados en los entornos de pruebas, garantizando la protección de la privacidad y la confidencialidad de los datos sensibles de los usuarios y de la organización. En su lugar, se deben utilizar datos enmascarados o ficticios que preserven la integridad de los datos y cumplan con las regulaciones de privacidad pertinentes.
* Los ambientes de desarrollo y producción deben protegerse mediante la aplicación de parches de seguridad y actualización de todas las herramientas de desarrollo, integración y pruebas, realizar configuraciones seguras de los diferentes ambientes, implementar controles de acceso a los mismos, gestión de cambios de los ambientes de desarrollo, pruebas y producción y la realización de copias de seguridad de los mismos.
* Se deben establecer segregación de roles y permisos en los diferentes ambientes, de tal forma que se impida que cualquier persona pueda realizar cambios a los ambientes sin revisiones ni aprobaciones previas.

## GESTIÓN DE CAMBIOS

* Se deben establecer y documentar procedimientos formales de cambios para sistemas nuevos o existentes, de tal forma que se garantice la confidencialidad, integridad y disponibilidad de la información durante todo el ciclo de vida de desarrollo, desde las etapas de diseño hasta los mantenimientos posteriores.
* La gestión de cambios debe incluir:
  + Lineamientos para la planificación y evaluación del impacto potencial de los cambios.
  + Autorización de cambios.
  + Comunicación de los cambios a las partes interesadas.
  + La realización de pruebas y los criterios de aceptación.
  + Planes de implementación de cambios.
  + Procedimientos de respaldo, cuando sea necesario.
  + Conservar registros de todo el proceso de cambios.
  + En caso que sea necesario modificar documentación existente debido a los cambios se debe actualizar.
* Es fundamental implementar un control riguroso de los cambios realizados en el ambiente de producción para prevenir cualquier impacto negativo en la integridad y disponibilidad de las aplicaciones. Este control debe incluir el seguimiento y la documentación detallada de todos los cambios realizados, así como la realización de pruebas exhaustivas antes de implementar dichos cambios en el ambiente de producción.
* Antes de implementar cambios en el ambiente productivo, es esencial realizar las mismas modificaciones en los ambientes de desarrollo y pruebas. Esta práctica asegura que los cambios sean probados y validados en entornos controlados antes de su implementación en el ambiente de producción.

# ANEXOS

* Ninguno.

# CONTROL DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Descripción del cambio** | **Autor** | **Fecha** |
| 01 | Creación del documento | Consultora del SGSI | 31/05/2024 |