# Políticas De desarrollo Seguro

MachineTrek

**Fecha de Emisión: 10/09/2024**
**Versión: 1**
**Fecha De Actualización: 10/09/2024**

María-José Aceituno
Josué Ávila
Carlos Kohler

# ÍNDICE

# 1. Objetivo General

El objetivo general de esta política es establecer lineamientos y prácticas para asegurar el desarrollo seguro de software, garantizando la integridad, confidencialidad y disponibilidad de la información.

# 2. Alcance

El alcance de esta política aplica para todos(as) los(as) funcionarios(as) de planta, contrata, honorarios y a toda persona natural o jurídica que preste servicios a la empresa y que, a raíz de ello, tenga la necesidad de realizar diversos accesos a los sistemas de información, físicos o lógicos, que la organización posea, incluyendo a los archivos de documentación, las aplicaciones, instalaciones, sistemas y redes.

Esta Política abarca todos los procesos operacionales, de apoyo y estratégicos que requieran en cualquiera de sus etapas la aplicación de controles de continuidad operacional.

Asimismo, abarca a todos los activos de información que la organización posea, de manera que la no inclusión explícita en el presente documento no constituye argumento para no proteger activos de información que se encuentren en otras formas, esta Política cubre toda la información impresa, almacenada electrónicamente, transmitida por correo o medios electrónicos, mostrada en películas o hablada en una conversación.

# 3. Documentos o marco de referencia.

Esta política se basa en los estándares de seguridad de la industria, incluyendo:
ISO/IEC 27001 Seguridad de la Información, ciberseguridad y protección de privacidad.
ISO/IEC 27002 / 17799 Política de Seguridad

# 4. Definiciones utilizadas

- **Desarrollo Interno:** Proyectos de software realizados por equipos internos de la empresa.
- **Desarrollo por terceros:** Proyectos realizados por contratistas o proveedores externos.
- **Gestión de Vulnerabilidades:** Proceso de identificación y mitigación de posibles vulnerabilidades en el software.
- **Activos de Información:** Todos aquellos elementos relevantes en la producción, emisión, almacenamiento, comunicación, visualización y recuperación de información de valor para la organización. Se constituye por:
    - La información propiamente tal, en sus múltiples formatos (papel; digital; texto; imagen; audio; video; transmisión verbal, etc.)
    - Los equipos, sistemas e infraestructura que soportan esta información.
    - Las personas que utilizan la información y que tienen el conocimiento de los procesos institucionales.

- **Código Malicioso:** El código malicioso es un tipo de código informático o script web dañino diseñado específicamente para crear y/o explotar vulnerabilidades en un sistema, permitiendo así la generación de puertas traseras, brechas de seguridad, robo de información y datos, así como otros perjuicios potenciales en archivos y sistemas de la información.
- **Confidencialidad:** Propiedad de la información que dispone a que esta no sea revelada a individuos, entidades o procesos no autorizados. (REF ISO 27000:2018).
- **Disponibilidad:** Propiedad de ser accesible y utilizable a pedido por una entidad autorizada. (REF ISO 27000:2018).
- **Integridad:** Propiedad de exactitud e integridad. (REF ISO 27000:2018).
- **Malware:** Cualquier tipo de software malicioso diseñado para dañar o explotar uno o más dispositivos, servicios o redes programables.
- **Negocio:** Bien o servicio prestado por una organización.
- **Política de Seguridad de la Información:** Conjunto de normas con el objetivo de proteger la información contra una amplia gama de amenazas para asegurar la continuidad del servicio y minimizar los daños, procurando la preservación de la confidencialidad, disponibilidad e integridad de la información.
- **Propietario de la Información:** Persona, natural o jurídica, responsable de la información y de los procesos que la manipulan, sean estos manuales, mecánicos o electrónicos. Debe participar activamente en la definición del valor de la información para el negocio, de manera de que se pueda definir los controles apropiados para protegerla.
- **Red:** Conexión entre equipos computacionales que permite compartir datos y recursos.
- **Riesgo de Seguridad de la Información:** Amenaza potencial que podría afectar activos de información, vinculados a los procesos de soporte institucional y/o a los procesos de provisión de productos estratégicos (bienes y servicios), establecidos en las definiciones estratégicas institucionales y, por tanto, causar daño a la organización.
- **Seguridad de la Información:** Preservación de la confidencialidad, integridad y disponibilidad de la información, (Ref ISO 27000:2018).
- **Software:** Producto intangible que permite a un equipo computacional desempeñar diversas tareas por medio de instrucciones lógicas, a través de diferentes tipos de programas.
- **Usuario:** Toda persona interna o externa que accede y utiliza activos de información institucionales.

# 5. Identificación de activos de información

- Datos del cliente (información personal, información financiera).
- Información de la empresa (estrategias, planes de negocio, propiedad intelectual).
- Datos de transacciones (historial de compras, detalles de pedidos, minutas de reparto).
- Infraestructura tecnológica (servidores, bases de datos, aplicaciones web, red de área local, computadoras y dispositivos móviles).
- Procesos críticos del negocio (procesamiento de pagos, gestión de inventario, onboarding y gestión de usuarios y clientes).

### Procesos de negocio

- Protección de la información del cliente.
- Gestión de incidentes de seguridad.
- Seguridad en transacciones financieras.
- Mantenimiento de la integridad de la plataforma web.
- Transporte de activos a destino.
- Verificación de estado de activos de cliente.
- Validación de identidad.
- Manejo de contratos de arrendamiento.

# 6. De la Política

## a. Consideraciones Generales:

Esta sección considera priorizar la seguridad en cada fase del ciclo de vida del desarrollo de software.

### Cultura de Seguridad:

Promover una cultura de seguridad en toda la organización, donde todos los miembros del equipo comprendan la importancia de la seguridad en el desarrollo de software. Esto con la finalidad de generar un ambiente donde se mantenga una estrecha relación entre usuario, empleado, prestador de servicios u otra entidad y el equipo de seguridad informática.

Incentivar la responsabilidad compartida, donde cada miembro del equipo contribuya a la seguridad desde su área de especialización. Sumado a esto se establece un protocolo de comunicación directa entre los miembros del equipo y el equipo de seguridad informática para discutir sobre posibles vectores de riesgo dentro del ambiente de desarrollo e implementación del software de la institución.

### Comunicación:

Establecer canales de comunicación efectivos entre los equipos de desarrollo y seguridad para compartir información sobre posibles amenazas y vulnerabilidades. Estos deben incluir canales de comunicación internos y externos de la organización, chat rooms y chat en vivo dentro de la organización, además de monitoreo continuo y automatizado de las comunicaciones de la empresa mediante el uso de lectura de palabras clave dentro de los medios de comunicación oficiales de la empresa.

## Evaluación Continua:

Implementar un enfoque de mejora continua, realizando revisiones periódicas de las políticas de desarrollo seguro para integrar las lecciones aprendidas y adaptarse a las evoluciones en las amenazas de seguridad.

Realizar evaluaciones de comportamiento digital de los empleados y colaboradores de la empresa con finalidad de identificar y reforzar conocimientos faltantes o deteriorados con respecto al desarrollo de software seguro.

## Conformidad con Estándares:

Asegurar que todas las actividades de desarrollo cumplan con los estándares de seguridad establecidos por la industria y la normativa aplicable.

Mantenerse informado sobre las mejores prácticas y estándares emergentes en seguridad informática para mejorar constantemente los procesos. Esto se llevará a cabo cada 6 (seis) meses o a medida que se modifiquen las normativas de seguridad vigentes.

## Reporte de Problemas:

Implementar un proceso claro para reportar problemas de seguridad, fomentando que todos los miembros del equipo informen de manera proactiva cualquier inquietud o incidente.

En caso de un incidente relacionado con hardware, malware o virus sospechoso, se deberá desconectar de inmediato el dispositivo de la red y apagarlo.

El incidente debe ser reportado al equipo de seguridad por correo electrónico lo antes posible, incluyendo la fecha, hora y un resumen de lo sucedido.

Asegurar que los informes de seguridad se manejen con confidencialidad y sin represalias, seguido de sesiones de capacitación para el personal involucrado.

## Integración con Desarrollo Ágil:

Integrar principios de seguridad en metodologías ágiles de desarrollo, asegurando que los equipos de seguridad estén alineados con los ciclos de desarrollo y entrega continua. Esto implica incorporar prácticas de seguridad desde las primeras fases de planificación, garantizando que los requisitos de seguridad estén presentes en las historias de usuario y en los criterios de aceptación. Además, los equipos de desarrollo deben recibir capacitación continua en seguridad, lo que les permitirá identificar y abordar posibles vulnerabilidades a lo largo del ciclo de vida del producto.

Ajustar las políticas según las necesidades específicas de los equipos ágiles, manteniendo la flexibilidad necesaria. Las políticas de seguridad no deben ser vistas como un obstáculo, sino como un facilitador para el desarrollo ágil, proporcionando directrices claras que permitan una

rápida adaptación a los cambios. Para esto, es esencial contar con puntos de revisión de seguridad en cada sprint, donde se evalúe la adherencia a las normativas de seguridad sin ralentizar el flujo de trabajo.

Integrar metodologías de validación de entregables en el pipeline de desarrollo ágil. Esto incluye automatizar pruebas de seguridad como parte de las pruebas de integración continua (CI) y entrega continua (CD), asegurando que cualquier vulnerabilidad sea detectada y corregida antes de que el código llegue a producción. Las herramientas de análisis estático y dinámico deben estar integradas en el pipeline, además de pruebas de penetración periódicas. De esta forma, se asegura que cada entrega cumpla con los estándares de seguridad necesarios, sin comprometer la velocidad ni la eficiencia del desarrollo ágil.

Además, promover la colaboración entre los equipos de desarrollo y seguridad es clave, permitiendo una comunicación constante para resolver posibles riesgos en tiempo real. Esto también puede lograrse mediante la adopción de prácticas como DevSecOps, que integra la seguridad como parte fundamental del ciclo de desarrollo, rompiendo silos y facilitando un enfoque más ágil y seguro.

### Evaluación de Riesgos:

Realizar evaluaciones de riesgos de manera regular para identificar nuevas amenazas y evaluar la efectividad de las medidas de seguridad existentes. Estas deben de realizarse semanalmente por defecto, los líderes de equipo o área pueden establecer aplazar esto hasta un periodo no mayor a 3 (tres) semanas del día originalmente asignado.

Incorporar la gestión de riesgos como parte integral de la toma de decisiones en el desarrollo de software. Se debe de generar un comité de gestión de riesgo en el departamento de desarrollo de la organización, con finalidad de entregar un veredicto y feedback dependiendo de las necesidades del proyecto y prioridad del software a desarrollar.

## b. Desarrollo por Terceros y Software Empaquetado:

### Evaluación de Proveedores:

Establecer un proceso de evaluación exhaustiva antes de contratar a terceros para el desarrollo de software. Se debe de establecer un perfil del tercero mediante una auditoría realizada por la organización y aprobada por el tercero con finalidad de determinar factores de riesgo con y para el mismo.

Evaluar la reputación y el historial de seguridad de los proveedores antes de comprometerse con proyectos.

## Cláusulas de Seguridad en Contratos:

Incorporar cláusulas específicas de seguridad en todos los contratos con proveedores externos de software, asegurando el cumplimiento de los más altos estándares de protección de la información. Se debe requerir la firma de acuerdos de confidencialidad y no competencia, garantizando la protección de los intereses estratégicos de la empresa.

Asimismo, es fundamental establecer y detallar los requisitos mínimos de seguridad que los proveedores deberán cumplir a lo largo de todo el ciclo de desarrollo. Estos requisitos deberán estar alineados con las normativas, legislaciones y certificaciones vigentes aplicables, tanto a nivel interno de la empresa como a las exigencias de los clientes, asegurando que todos los productos y servicios adquiridos se ajusten a los estándares de seguridad y calidad requeridos.

## Análisis de Riesgos:

Realizar un análisis de riesgos detallado para cada proyecto que involucre a terceros, identificando de manera proactiva las posibles amenazas y vulnerabilidades que puedan comprometer la seguridad de la empresa. Este análisis debe contemplar tanto los aspectos técnicos como los operacionales del proyecto.

Es recomendable contar con un equipo externo especializado en pruebas de penetración (pentesting), responsable de realizar una evaluación exhaustiva de las vulnerabilidades de la empresa, así como de detectar posibles fallas dentro de la cadena de desarrollo. Este equipo externo debe operar de manera independiente para garantizar un enfoque imparcial y profundo en la identificación de riesgos.

El individuo o entidad encargada de llevar a cabo dicho análisis deberá firmar acuerdos de confidencialidad estrictos, asegurando la protección de la información sensible de la empresa. Además, será responsable de proporcionar una completa transparencia en la presentación de los datos obtenidos durante la evaluación de vulnerabilidades. Este proceso culminará en la entrega de un informe detallado, en el que se presentarán las vulnerabilidades detectadas junto con recomendaciones claras y viables para su mitigación.

## Revisión de Código Externo:

Exigir la revisión y auditoría exhaustiva del código fuente de cualquier software proporcionado por terceros antes de su implementación en los sistemas de la empresa. Este proceso debe incluir una evaluación detallada para identificar la presencia de vulnerabilidades conocidas y asegurar que se sigan prácticas de codificación segura.

Además, la auditoría debe garantizar que el software cumpla con los estándares de seguridad establecidos por la empresa, incluyendo normativas regulatorias y mejores prácticas de la industria. Cualquier vulnerabilidad detectada deberá ser corregida antes de la implementación final, mitigando así posibles riesgos que puedan comprometer la integridad y seguridad del sistema.

## Actualizaciones y Parches:

Establecer un proceso claro para recibir y aplicar actualizaciones y parches de seguridad proporcionados por terceros proveedores. Favorecer la estabilidad de los sistemas de la empresa. Se deben de consultar con comité de desarrollo seguro y contar con respaldo o posibilidad de realizar rollback de sistemas si fuera necesario tras instalación de actualizaciones de seguridad con la finalidad de establecer integridad de los datos de la empresa.

Mantenerse informado sobre las actualizaciones de seguridad de las soluciones de software empaquetado utilizadas. Deshabilitar actualizaciones de usuario, establecer actualizaciones diarias de sistemas de la empresa y terminales de usuario.

## Integración Segura:

Garantizar que la integración de software de terceros se realice de manera segura, evitando el uso de configuraciones predeterminadas que puedan ser inseguras, y habilitando únicamente las funciones estrictamente necesarias para su funcionamiento. Este enfoque minimiza la superficie de ataque y reduce el riesgo de posibles vulnerabilidades.

Es fundamental validar todas las interfaces y conexiones utilizadas por el software para prevenir posibles puntos de acceso no autorizados o vulnerabilidades. Además, se debe asegurar que las credenciales, llaves de acceso y cualquier otra información sensible no estén referenciadas en texto plano dentro del código fuente. Para ello, se deben emplear prácticas como el cifrado de datos sensibles y la utilización de gestores de secretos para garantizar la protección de la información crítica.

## Gestión de Acceso:

Aplicar principios de gestión de acceso para restringir el acceso a las funciones y datos sólo a aquellos que sean necesarios para el funcionamiento del software.
Establecer criterios de whitelisting. Otorgar solo permisos de acceso requeridos para funciones del usuario del sistema de la empresa y funciones a desarrollar en dicho sistema.
Revocar inmediatamente los accesos de terceros al completar el desarrollo.

## Auditorías de Seguridad:

Realizar auditorías de seguridad periódicas en el software desarrollado por terceros, verificando el cumplimiento continuo de los estándares de seguridad.
Incorporar cláusulas contractuales que permitan auditorías de seguridad por parte de la organización.

### Capacitación y Concientización:

Proporcionar formación en seguridad a los equipos de desarrollo de terceros, asegurándose de que estén al tanto de las políticas y estándares de seguridad de la empresa.
Incluir requisitos de seguridad en la documentación y capacitación proporcionada a los terceros.

## c. Desarrollos Internos:

En este segmento se establecerán las políticas de seguridad respecto al desarrollo interno de la compañía, mediante esto se ven afectados los procesos de desarrollo realizado por los diferentes equipos de desarrollo de la empresa, a su vez de los sistemas involucrados en dicho desarrollo, entre estos puntos se encuentran los siguientes:

### Revisiones de Código:

Establecer un proceso regular de revisiones de código entre pares para identificar y corregir posibles vulnerabilidades dentro de cada equipo de trabajo. Esto debe de establecerse de manera predeterminada en un periodo semanal. Este periodo de revisión periódica no puede ser menor a 3 (tres) días hábiles y no puede separar las 3 (tres) semanas como plazo máximo entre revisiones.

Incluir criterios de seguridad como parte integral de las revisiones de código. Esto refiere a realizar énfasis en reforzar los estándares de seguridad de la empresa, revisar que no se codifican valores en duro y reforzar uso de variables de entorno dentro del desarrollo y mantenimiento del software y plataformas de la empresa.

### Formación Continua en Seguridad:

Proporcionar formación regular sobre las últimas amenazas de seguridad y mejores prácticas de desarrollo seguro. Esto debe realizarse mediante medios de comunicación oficiales de la empresa, a su vez se debe de establecer protocolos de entrenamiento de personal tras incidentes que involucren a su departamento o equipo de desarrollo.

Asegurar que los equipos internos estén al tanto de las políticas de seguridad y procedimientos establecidos.

### Pruebas de Seguridad Automatizadas:

Integrar herramientas de análisis estático de código y pruebas de seguridad automatizadas en los procesos de desarrollo.
Configurar alertas para identificar posibles vulnerabilidades durante la fase de desarrollo.

## Gestión de Identidades y Accesos:

Implementar principios de gestión de identidades y accesos para limitar el acceso a los recursos y datos según sea necesario.
Revocar inmediatamente los privilegios de acceso cuando un empleado cambie de función o deje la organización.

## Desarrollo Seguro desde el Inicio:

Integrar la seguridad desde el inicio del ciclo de vida del desarrollo de software.
Incorporar la gestión de riesgos y la evaluación de amenazas en las fases iniciales del desarrollo.

## Control de Versiones Seguro:

Establecer prácticas seguras para el control de versiones y la gestión de ramas, asegurando la integridad y la trazabilidad del código.
Restringir el acceso a los repositorios de código a equipos autorizados.
Establecer scripts de validación de CI/CD y seguridad dentro de pipelines de trabajo y producción.

## Gestión de Dependencias:

Validar y mantener actualizadas las bibliotecas y dependencias del software para abordar vulnerabilidades conocidas.
Establecer un proceso para evaluar la seguridad de nuevas dependencias antes de su implementación.

## Gestión de Cambios Controlada:

Implementar un proceso de gestión de cambios que garantice la revisión y aprobación antes de la implementación.
Registrar y auditar los cambios realizados en el código y la configuración.

## Evaluación de Desempeño Segura:

Integrar la seguridad en las evaluaciones de desempeño del equipo de desarrollo.
Reconocer y recompensar las prácticas seguras en el desarrollo de software.

## Investigación y Desarrollo Seguro:

Aplicar principios de seguridad incluso en proyectos de investigación y desarrollo, asegurando que los nuevos desarrollos cumplan con los estándares de seguridad establecidos.

## d. Gestión de Vulnerabilidades:

En el siguiente segmento se especificará el proceso de gestión de vulnerabilidades dentro del proyecto de desarrollo y procesos integrales de la empresa.

### Evaluación Continua:

Implementar evaluaciones continuas de vulnerabilidades a lo largo del ciclo de vida del desarrollo de software mediante herramientas automáticas y análisis estático de código. Estas evaluaciones deben de especificar qué entorno se ve afectado, qué componentes del software son lo que se ven afectados y qué paquetes o

### Clasificación de Vulnerabilidades:

Establecer una clasificación de vulnerabilidades según su gravedad y riesgo potencial, con criterios claros para priorizar y abordar las vulnerabilidades de manera adecuada.
Se deben de utilizar estándares de clasificación como lo son CVE e identificación mediante hash MD5 de dicha vulnerabilidad a la vez de niveles de 0 a 10 en cuanto al peligro que conlleva la vulnerabilidad.

### Gestión de Parches:

Desarrollar un proceso eficiente para la gestión de parches que permita la aplicación rápida de soluciones a las vulnerabilidades identificadas, asegurando su implementación oportuna y efectiva. Contar con nightly builds del software con actualizaciones de seguridad actualizadas, a la vez de actualizaciones de seguridad semanales.

### Colaboración y Comunidad:

Colaborar con la comunidad de seguridad para compartir información sobre vulnerabilidades y participar en programas de recompensas por errores cuando sea apropiado.

Generar un proceso de recompensas interno de vulnerabilidades, este podrá ser concursable solo por externos a la organización o miembros de departamentos diferentes. Se debe de contar con transparencia de vulnerabilidades y sancionar las vulnerabilidades incorporadas intencionalmente en el codebase de la empresa por colaboradores de la organización.

### Formación y Concientización:

Proporcionar formación periódica a los equipos de desarrollo sobre la gestión efectiva de vulnerabilidades, garantizando su familiaridad con los procesos y herramientas utilizados.

### Informes de Vulnerabilidades:

Establecer un canal claro y seguro para informar sobre vulnerabilidades descubiertas, asegurando la confidencialidad y tomando medidas rápidas para abordarlas.

Estos informes de vulnerabilidades deben de entregarse impresos o mediante medios digitales al encargado de recepción de informes de seguridad mediante correo electrónico.

### Integración con Desarrollo Ágil:

Integrar la gestión de vulnerabilidades en los ciclos de desarrollo ágil, asegurándose de que las evaluaciones de seguridad sean parte integral de cada iteración.

### Registro de Vulnerabilidades:

Mantener un registro detallado de todas las vulnerabilidades identificadas, las acciones tomadas y las fechas de resolución, utilizando esta información para análisis de tendencias y mejora continua.

## e. Documentación:

Documentar los procesos de desarrollo, incluyendo medidas de seguridad implementadas.
Mantener registros de auditoría y documentación técnica actualizada.

Toda funcionalidad incorporada al codebase de la empresa debe de ser registrada mediante documentación pertinente a la misma, esto debe de contener un desglose de arquitectura de software y estructuras de datos a utilizar, además de explicación de funciones y objetos internos.

Todo repositorio debe de contar con documentación ya sea en formato informe o mediante un archivo READ_ME.md en la carpeta raíz del repositorio.

## f. Evaluaciones o Casos de Negocios:

Se debe de evaluar el impacto de la seguridad en los casos de negocio constantemente en el proceso de desarrollo de software de la empresa. Se debe de asignar una puntuación de 0 a 10 con finalidad de determinar el impacto de la implementación de un nuevo sistema o software al modelo de negocio de la empresa.

Integrar análisis de riesgos en la planificación del desarrollo, se debe de mantener actualizado el impacto y riesgos que conllevan los proyectos de software de la empresa y como su desarrollo puede afectar las vulnerabilidades de la misma.

## g. Especificación Detallada de los Requerimientos:

Incluir requisitos de seguridad en la fase de especificación de requisitos, estos deben incluir normativas, leyes, certificaciones y otros requerimientos de seguridad de proyecto.

Validar y verificar los requisitos de seguridad durante el desarrollo, esto significa mantener una actualización constante del alcance del proyecto y de seguridad del proyecto.

## h. Diseño del Sistema e Interfaces:

Incorporar la seguridad en el diseño arquitectónico del sistema, establecer validaciones dentro de la arquitectura del sistema de la organización y de software a desarrollar, contar con respaldo y posibilidad de rollback en caso de ser necesario.

## i. Codificación y Pruebas:

Aplicar buenas prácticas de codificación segura, no almacenar credenciales en texto plano o comentadas dentro del código desarrollado, establecer normativas de seguridad, criptografía y cifrado.

Realizar pruebas de seguridad durante las fases de desarrollo, integración, pruebas, despliegue, implementación y mantenimiento.

## j. Implementación y Pruebas:

Realizar implementaciones seguras según las mejores prácticas, se debe de realizar pruebas y asignación de puertos correspondientes a las tecnologías utilizadas y requerimientos del proyecto.

Ejecutar pruebas de seguridad antes de la puesta en producción, corregir las vulnerabilidades sobre nivel 3 antes de ser puesto el software en producción.

## k. Post Implementación y Marcha Blanca:

Monitorear activamente el software después de la implementación. Generar pruebas de penetración semanales y contratar a equipos externos con finalidad de encontrar vulnerabilidades en software alojado en ambiente de producción.
Realizar una fase de marcha blanca para validar la seguridad post-implementación.

## l. Controles por Implementar:

Mantener una lista actualizada de controles de seguridad pendientes de implementar.
Establecer un plan para la implementación gradual de controles de seguridad.

# 7. Métodos y mecanismos de distribución

- Distribuir esta política a todos los miembros del equipo de desarrollo.
- Proporcionar formación sobre la política de desarrollo seguro.

# 8. Periodo de revisión del documento

Este documento será revisado semestralmente o ante cambios significativos en los procesos de desarrollo.
Después de cualquier cambio de leyes o normativas que afecten a la seguridad de información y la privacidad de los datos, se revisará y se ajustará para calzar con las leyes vigentes.
Ante posibles errores se realizará una copia de seguridad y se archivarán cada versión cambiada, esto dará paso a un historial de versiones.

# 9. Control de cambios

Cualquier cambio en esta política debe ser aprobado por el equipo informático y documentado en el registro de cambios.

Esta política entrará en vigor a partir del 10 de Septiembre de 2024.

| Versión | Fecha | Puntos modificados | Resumen de las modificaciones |
|---|---|---|---|
| 1.0 | 10/09/2024 | Todos los puntos | Creación del documento. |