	Política de Desarrollo de Software Seguro		
Código: P08	Versión: 001		

1. Objetivo y Alcance

1.1. Objetivo

Establecer los requisitos y las actividades de seguridad que deben integrarse en cada fase del Ciclo de Vida de Desarrollo de Software (SDLC) para garantizar que las aplicaciones sean seguras desde su diseño (Security by Design).

1.2. Alcance

Aplica a:

- Todos los sistemas de información, aplicaciones y componentes de software desarrollados, adquiridos o modificados por AOS para nuestros clientes.
- Todos los desarrolladores, arquitectos, analistas y personal involucrado en el proceso de desarrollo.

2. Integración de la Seguridad en el SDLC

La seguridad debe ser un proceso continuo, no una etapa final. Los siguientes controles deben integrarse en las fases del SDLC:

Fase del SDLC	Requisito de Seguridad	Evidencia y Control
1. Requisitos y Diseño	a. Modelado de Amenazas: Identificar, clasificar y documentar los posibles ataques (riesgos) contra la arquitectura del sistema antes de comenzar el código. b. Definición de Requisitos: Incluir requisitos de seguridad explícitos (ej. "La aplicación debe usar MFA", "Los datos sensibles deben cifrarse en reposo").	Documento de Modelado de Amenazas. Lista de Chequeo de Requisitos de Seguridad.
2. Construcción y Codificación	a. Estándares de Codificación Segura: Obligatorio seguir	Guía interna de Estándares de Codificación Segura.

Fase del SDLC	Requisito de Seguridad	Evidencia y Control
	estándares de codificación segura (ej. OWASP Top 10, CWE). b. Ofuscación/Cifrado de Código: Aplicar técnicas de cifrado o ofuscación a las porciones críticas del código fuente. c. No Credenciales en Código: Prohibición estricta de almacenar contraseñas, tokens o claves de API en el código fuente.	Revisión de código por pares.
3. Pruebas y Aseguramiento de Calidad (QA)	a. Análisis Estático (SAST): Ejecutar herramientas de Análisis Estático de Código sobre el código fuente antes de la compilación para identificar fallos de seguridad. b. Análisis Dinámico (DAST): Ejecutar pruebas automatizadas en la aplicación en un entorno de pruebas, simulando ataques. c. Pruebas de Penetración (Pen Test): Programar Pen Tests formales antes del pase a producción.	Informe de Resultados SAST/DAST (ver sección 4). Informe de Pruebas de Penetración.
4. Despliegue y Producción	a. Revisión de Configuración: Verificar que los ajustes del entorno de producción (servidores web, bases de datos) cumplan con los estándares de hardening definidos.	Lista de Chequeo de Hardening de Producción. Registro de Aprobación del Release.

Fase del SDLC	Requisito de Seguridad	Evidencia y Control
	b. Control de Release: Asegurar que solo el código revisado y aprobado por seguridad sea desplegado.	

3. Uso de Herramientas de Análisis Automatizado

3.1. Requerimiento de Herramientas SAST y DAST

1. Herramientas SAST (Static Application Security Testing): Se debe implementar y utilizar una herramienta SAST de forma rutinaria sobre el código para identificar vulnerabilidades comunes (ej. Inyección SQL, Cross-Site Scripting - XSS) sin necesidad de ejecutar la aplicación.
2. Herramientas DAST (Dynamic Application Security Testing): Se debe utilizar una herramienta DAST para probar la aplicación mientras está en ejecución y simular ataques de un hacker.

3.2. Gestión de Hallazgos de Herramientas

- Mitigación Obligatoria: Todos los hallazgos de seguridad clasificados como Críticos y Altos por las herramientas SAST/DAST deben ser mitigados antes del despliegue a Producción.
- Falsos Positivos: La justificación de un falso positivo debe ser documentada y aprobada por el Oficial de Seguridad.

4. Evidencia de Implementación

4.1. Uso de Herramientas SAST/DAST

El equipo de desarrollo debe mantener evidencia del uso de herramientas de análisis de seguridad, que puede ser:

- Informes de Resultados: Copia de los informes generados por la herramienta (SAST/DAST) para cada release o iteración importante.
- Integración en CI/CD: Captura de pantalla o registro que muestre la integración de la herramienta en el pipeline de Integración/Entrega Continua (CI/CD).

4.2. Lista de Chequeo de Revisión de Código (Manual)

En caso de que las herramientas automatizadas sean insuficientes o no estén implementadas, el equipo debe aplicar una Lista de Chequeo de Revisión de Código Manual para cada componente crítico. Esta lista debe incluir:

ID	Control de Seguridad	Cumple (Sí/No)	Comentarios/Mitigación
DCS-01	¿Los inputs de usuario están validados y sanitizados para prevenir Inyección SQL/XSS?		
DCS-02	¿Se utiliza hashing y salt para almacenar contraseñas de usuarios?		
DCS-03	¿Se han eliminado las credenciales codificadas (hardcoded) en el código fuente o en archivos de configuración?		
DCS-04	¿Se ha implementado el control de acceso a nivel de función (autorización)?		
DCS-05	¿Se aplica un mecanismo de logging y manejo de errores seguro (no divulgar stack traces o información sensible)?		