0INDICE

1. OBJETIVO

El objetivo del presente documento es Definir los principios fundamentales de seguridad que deben aplicarse durante el diseño, desarrollo e implementación de sistemas y aplicaciones, con el fin de asegurar que la seguridad sea considerada desde la concepción (“security by design”) y durante todo el ciclo de vida del software.

2. ALCANCE

Este documento aplica a todos los proyectos de desarrollo de software, arquitecturas técnicas, infraestructuras, API, y servicios desarrollados o mantenidos por DACTA SAC.
Incluye sistemas internos y soluciones para clientes.

3. RESPONSABLES

- **Gerente de Proyectos**
  Aprobar los principios de diseño seguro y asegurar su aplicación
- **Coordinador de Desarrollo**
  Aplicar estos principios en el diseño de soluciones
- **Desarrolladores**
- Cumplir las directrices de codificación segura y arquitectura establecidas.
- **OSI**
  Validar la adecuación de los principios y realizar revisiones periódicas.

4. CONTENIDO

Los siguientes principios deberán aplicarse en todo diseño técnico y revisión de arquitectura:

4.1 Seguridad por diseño y por defecto

La seguridad debe incorporarse desde las s fases iniciales del proyecto, no como algo añadido al final.
La seguridad por diseño implica que todos los componentes, interfaces y flujos de datos son evaluados bajo criterios de confidencialidad, integridad, disponibilidad y trazabilidad.
Las configuraciones por defecto deben ser seguras (mínimos privilegios, contraseñas fuertes, servicios innecesarios deshabilitados)

4.2 Principio de mínimo privilegio

Cada componente, servicio o usuario debe operar con los privilegios mínimos necesarios.
El principio establece que cada usuario, proceso, sistema o servicio solo debe tener los permisos estrictamente necesarios para realizar su función.
Tiene como finalidad reducir el impacto de errores humanos, vulnerabilidades o ataques, porque incluso si una cuenta o servicio es comprometido, los daños estarán limitados.

4.3 Defensa en profundidad

Implementar capas sucesivas de seguridad (firewalls, autenticación, validación de datos, cifrado, auditoría) de modo que si una capa falla, las demás sigan protegiendo la información.

4.4 Separación de funciones

Separar los entornos de desarrollo, pruebas y producción arquitectura o diseño.
Evitar que una misma cuenta o proceso tenga permisos para tareas incompatibles (por ejemplo, desarrollo y despliegue)

4.5 Gestión segura de datos y secretos

Establece que toda la información sensible como claves, contraseñas, tokens, certificados, configuraciones críticas o datos personales, deberán protegerse mediante almacenamiento seguro, evitando exposición accidental o accesos no autorizados.
Cifrar datos en tránsito (TLS) y en reposo (AES-256 o superior).

4.6 Validación y sanitización de entradas

Este principio establece que toda la información que ingresa al sistema debe ser considerada potencialmente maliciosa o incorrecta, y por tanto, debe validarse y limpiarse antes de ser procesada o almacenada.
Validar todas las entradas de usuario o de fuentes externas para prevenir ataques de inyección o desbordamiento.

4.7 Gestión de dependencias y librerías

Establece que todas las dependencias, librerías y frameworks de terceros utilizados en el desarrollo deben ser, verificados, actualizados y controlados, deben de provenientes de fuentes confiables, además deberán ser monitoreados ante vulnerabilidades conocidas.
Usar únicamente librerías aprobadas, mantener actualizadas las dependencias y gestionar vulnerabilidades conocidas (CVE).

4.8 Registro y auditoría

Establece que todo sistema debe generar, almacenar y proteger registros de eventos relevantes de seguridad y operación, de manera que permitan:

- Detectar comportamientos anómalos o incidentes,
- Investigar la causa raíz de eventos, y
- Asegurar la trazabilidad y responsabilidad de las acciones realizadas

Los logs se almacenan en repositorios con acceso restringido únicamente al personal autorizado. Estos registros permiten detectar incidentes, investigar y corregir rápidamente.

4.9 Resiliencia y disponibilidad

Este principio establece que los sistemas deben mantener su funcionamiento incluso ante fallos, interrupciones o ataques, y que deben poder recuperarse rápidamente para minimizar el impacto operativo.
Incluir copias de seguridad y mecanismos de redundancia.

4.10 Privacidad y cumplimiento

Integrar la protección de datos personales y el cumplimiento legal en todas las fases de diseño, desarrollo y operación de sus sistemas, conforme a los principios de privacidad por diseño y por defecto

5 PROCESO DE REVISIÓN DE DISEÑO SEGURO

5.1 Inicio del proyecto

El equipo define la arquitectura considerando los principios anteriores

5.2 Revisión técnica

El Coordinador de Desarrollo debe validar el cumplimiento de los principios mediante una Checklist.

5.3 Aprobación

No se avanza al desarrollo sin la aprobación formal de la revisión

5.4 Registro

Las revisiones y evidencias se documentan en el repositorio del proyecto o sistema de gestión de calidad

6 DOCUMENTACIÓN Y EVIDENCIA

- Checklist de revisión de arquitectura segura
- Informes de evaluación de riesgos técnicos
- Evidencias de aprobación por parte del responsable técnico o de seguridad.

7 REVISIÓN Y MEJORA CONTINUA

Este documento será revisado anualmente o cuando se introduzcan cambios significativos en la tecnología o metodologías de desarrollo

Cualquier mejora o ajuste deberá ser aprobado por la Gerencia de proyectos y/o el Responsable de Seguridad

8 REFERENCIAS

- ISO/IEC 27001:2022, control A.8.27
- ISO/IEC 27002:2022, cláusula 8.27
- OWASP Software Assurance Maturity Model (SAMM)
- NIST SP 800-218 – Secure Software Development Framework (SSDF)
- OWASP Top 10

9 CONTROL DE CAMBIOS

Versión	Fecha	Descripción	Elaborado por	Revisado por	Aprobado por
1	22/10/2025	Emisión	**Coordinador de Desarrollo**	**Gerente de Proyectos**	**Gerente General**
Firmas de la versión vigente					

	A. Morales	S. Rafaile	G. Rafaile

Identificación de las modificaciones

Versión	Descripción de cambios