CONTROL: 8.28 ARCHIVOS DE ORIGEN DE SOFTWARE

Proyecto: Aplicación Móvil "Alerta Mujer" Fecha: 2 de octubre de 2025 Estado del Control en el Proyecto: Faltante (F)

1. Propósito

El control 8.28 exige la protección y gestión controlada de los archivos de código fuente y otros archivos de desarrollo (ej. claves, librerías, *assets*) para prevenir el acceso no autorizado, la modificación maliciosa, o la pérdida. La pérdida de este control podría resultar en la filtración de la lógica de negocio, la explotación de vulnerabilidades ocultas o la incapacidad de reconstruir la aplicación en caso de desastre.

Para "Alerta Mujer", la aplicación de este control es vital para proteger los secretos del sistema, como los algoritmos de cifrado (5.32) y la lógica de activación discreta (RF10).

2. Procedimiento de Aplicación del Control 8.28

La gestión de los archivos de origen se basará en el uso de un Sistema de Control de Versiones (VCS) centralizado y rigurosamente configurado (ej. Git/GitHub Enterprise, GitLab, Azure DevOps).

2.1. Control de Acceso Riguroso

El acceso al repositorio de código fuente debe gestionarse según el principio de Mínimo Privilegio.

- Identificación: Se utilizarán las identidades únicas del personal de desarrollo. No se permitirán cuentas compartidas.
- Autorización:
    - El acceso de lectura (*Read*) se concederá a todo el equipo de desarrollo.
    - El acceso de escritura (*Write*) solo se concederá a desarrolladores principales (*Core Developers*) y administradores del repositorio.
    - El acceso a las ramas de producción (*main/master*) solo se permitirá a través de solicitudes de combinación (*Merge Requests*) aprobadas (Ver 2.3).
- Administración de Repositorio: El acceso a la configuración del repositorio (ej. gestión de claves de despliegue) estará restringido únicamente al equipo DevOps/Administración de Seguridad.

2.2. Versionamiento y Trazabilidad (Control de Integridad)

Todo cambio en el código fuente debe ser rastreado y reversible, asegurando la integridad del sistema.

- Política de *Commits*: Cada cambio debe documentarse con un mensaje de *commit* que explique el propósito del cambio y vincule a un *ticket* o requisito.
- Etiquetado (*Tagging*): Las versiones estables que se despliegan a producción deben ser etiquetadas (ej. v1.0.0) para permitir una fácil identificación y restauración (*rollback*) de una versión de código específica.
- Revisiones de Código (*Code Review*): Se implementará la obligatoriedad de que todo cambio de código sea revisado y aprobado por al menos un desarrollador par antes de ser fusionado a la rama principal.

2.3. Protección de Información Sensible en Repositorios

Se tomarán medidas para evitar que información crítica se almacene directamente en el repositorio de código.

- Manejo de Secretos: Las claves de API, *tokens* o credenciales sensibles (ej. para el servicio de SMS) NO se almacenarán en texto plano en el repositorio. Se utilizarán herramientas de gestión de secretos (ej. HashiCorp Vault, AWS Secrets Manager) o variables de entorno gestionadas por el sistema CI/CD.
- Búsqueda Preventiva: Se implementarán herramientas de escaneo (*pre-commit hooks* o análisis CI/CD) para detectar y bloquear la subida de patrones de texto que parezcan ser credenciales antes de que lleguen al repositorio.

2.4. Respaldo y Retención

El repositorio, al ser un activo crítico (5.9 faltante), debe ser protegido contra fallos del proveedor.

- Frecuencia de Respaldo: El repositorio completo será respaldado fuera de la plataforma principal del VCS (ej. en un almacenamiento *cold*) al menos una vez por semana.
- Retención: Los respaldos completos del código fuente de cada versión mayor (*Release*) se retendrán durante un mínimo de 5 años.

LIDER DEL PROYECTO. EQUIPO DE TRABAJO. EQUIPO DE TRABAJO.