Requisitos y recomendaciones de diseño seguro en aplicación web de gestión PyME

Versión 1.0
01/2026

Proyecto: Mini SSDLC aplicado a sistemas web

Autor: Pedro Garvich

Índice

1. Introducción

Este documento define un conjunto de requisitos y recomendaciones de diseño seguro para un sistema de gestión web interno, elaborados **previo a la implementación del código**, como parte de un enfoque de *Secure Software Development Life Cycle (SSDLC)* con orientación *shift-left*.

Las recomendaciones aquí presentadas se derivan directamente de el documento de contexto del sistema (que define el alcance funcional, los roles y los activos críticos) y el modelado de amenazas realizado mediante la metodología **STRIDE**, aplicado a los principales flujos y componentes del sistema.

El objetivo de este artefacto es **traducir los riesgos identificados en el modelado de amenazas en decisiones concretas de diseño**, proporcionando una guía clara para:

- la implementación de controles de seguridad durante el desarrollo;
- la definición de criterios verificables mediante análisis estático de código (SAST);
- y la planificación de escenarios de prueba relevantes para análisis dinámico (DAST);

El alcance del documento se limita a la **definición de requisitos no funcionales de seguridad a nivel de aplicación**, incluyendo aspectos como autenticación, autorización, gestión y protección de datos, comunicaciones seguras y controles de auditoría.
No se aborda la arquitectura final del sistema ni su implementación concreta, sino que se establecen **criterios de diseño y controles esperados** que deben ser considerados durante el desarrollo.

Quedan fuera de alcance actividades como auditorías de infraestructura, pruebas de penetración activas, revisiones exhaustivas de código fuente, evaluaciones de cumplimiento normativo o la gestión operativa de entornos productivos.

Las recomendaciones se encuentran alineadas con buenas prácticas reconocidas (OWASP ASVS, OWASP Proactive Controls y CWE), aplicadas a un nivel acorde a un **sistema interno de gestión y a un proyecto de alcance acotado**.

Asimismo, los requisitos se organizan por dominios de riesgo (autenticación, autorización, gestión de datos, auditoría, entre otros), con el fin de facilitar su adopción práctica por parte de desarrolladores y su posterior validación mediante actividades de testing de seguridad.

2. Principios generales de diseño seguro

Los requisitos y recomendaciones definidos en este documento se sustentan en un conjunto de **principios generales de diseño seguro**, que orientan las decisiones técnicas antes y durante la implementación del sistema.

Estos principios no constituyen controles específicos, sino **criterios transversales** que deben aplicarse de forma consistente en toda la solución.

Principio de mínimo privilegio

Cada usuario y componente del sistema deberá contar únicamente con los permisos estrictamente necesarios para cumplir su función.

En particular, las operaciones de alto impacto sobre datos críticos deben estar restringidas y claramente diferenciadas de las tareas operativas cotidianas.

Separación de responsabilidades y roles

Los roles funcionales del sistema (socios, empleados y administrador del sistema) deben reflejarse en controles de acceso explícitos, evitando la concentración innecesaria de privilegios en un único tipo de usuario.

Las tareas técnicas de administración global se consideran excepcionales y separadas del uso normal del sistema.

Defensa en profundidad

Los controles de seguridad no deben depender de un único mecanismo. La protección de activos críticos debe apoyarse en múltiples capas, incluyendo validaciones de entrada, control de acceso, gestión de sesiones, registros de auditoría y mecanismos de recuperación ante fallos.

Seguridad por defecto (*secure by default*)

Las configuraciones iniciales del sistema deben ser restrictivas, reduciendo la superficie de ataque y evitando comportamientos inseguros por omisión.

El acceso a funcionalidades sensibles debe requerir una habilitación explícita.

Trazabilidad y no repudio

Las operaciones relevantes sobre datos críticos deben quedar registradas de forma confiable, permitiendo reconstruir acciones realizadas por los usuarios y reduciendo el riesgo de repudio ante incidentes o errores operativos.

Fallo seguro (*fail secure*)

Ante errores, excepciones o condiciones inesperadas, el sistema debe mantener un estado seguro, evitando la exposición de información sensible o la ejecución de acciones no autorizadas.

3. Requisitos de diseño por dominio de riesgo

3.1 Autenticación y gestión de sesiones

3.1.1 Objetivo de seguridad

Garantizar que los usuarios del sistema sean **correctamente identificados y autenticados**, y que las sesiones activas no puedan ser suplantadas, reutilizadas o elevadas de forma indebida, especialmente en el caso de roles con acceso a información financiera y operaciones críticas (socios y administrador).

3.1.2 Requisitos y recomendaciones de diseño

REQ-AUTH-01 – Autenticación individual y no compartida

Cada usuario deberá contar con credenciales únicas e individuales.
No se permitirá el uso de cuentas compartidas entre múltiples personas.

REQ-AUTH-02 – Gestión segura de credenciales

Las contraseñas deberán almacenarse utilizando mecanismos de hash robustos y con *salt*. Ejemplos recomendados: bcrypt, argon2.

No deberán almacenarse ni transmitirse contraseñas en texto plano.

REQ-AUTH-03 – Protección contra suplantación de sesión

El identificador de sesión deberá:

- Ser generado de forma segura. Recomendado utilizar urandom, CryptGenRandom (128 bits mínimo).
- Rotarse luego de la autenticación.
- Invalidarse correctamente al cerrar sesión. Destruír sesión en el backend, invalidar en almacenamiento de sesiones, rotar id al logout.

REQ-AUTH-04 – Expiración y control de sesiones

Las sesiones deberán expirar tras un período razonable de inactividad.
El sistema deberá impedir el uso de sesiones caducadas o invalidadas.

REQ-AUTH-05 – Autenticación reforzada para roles sensibles

Los roles con acceso a información financiera consolidada y operaciones críticas (socios) deberán contar con mecanismos de autenticación reforzada, tales como:

- Doble factor de autenticación. Recomendado el uso de mecanismos TOTP (ejemplo: Google Authenticator).
- Al menos un control de reautenticación ante operaciones críticas en sesión (borrado de cargas, lectura de reporte completo mensual).

REQ-AUTH-06 – Protección frente a fuerza bruta y abuso de autenticación

El sistema deberá implementar controles para limitar intentos repetidos de

autenticación fallida, reduciendo el riesgo de ataques automatizados. El umbral exacto deberá ajustarse según el perfil de riesgo del sistema.

Desbloquear la cuenta excepcional de administrador requeriría una medida adicional como la aprobación de todos los socios.

REQ-AUTH-07 - Política de contraseñas

Las contraseñas de los usuarios deberán cumplir con una política alineada a buenas prácticas actuales, priorizando la resistencia a ataques de fuerza bruta, credential stuffing y reutilización de credenciales, sin imponer requisitos de complejidad arbitrarios.

En particular:

- La longitud mínima de las contraseñas deberá ser:
  - **≥ 8 caracteres** para empleados.
  - **≥ 15 caracteres** para roles sensibles (socios y administrador).
- No se deberán imponer reglas obligatorias de composición (mayúsculas, símbolos, números) que degraden la usabilidad sin aportar seguridad efectiva.
- No se deberá exigir rotación periódica de contraseñas salvo ante:
  - Indicios de compromiso.
  - Incidentes de seguridad.
  - Restablecimiento forzado por administración.
- El sistema deberá prevenir el uso de contraseñas débiles o ampliamente conocidas mediante:
  - Listas de contraseñas prohibidas.
  - Bloqueo de contraseñas presentes en diccionarios o filtraciones conocidas.
- Las contraseñas deberán validarse únicamente del lado servidor.

3.1.3 Trazabilidad con amenazas identificadas

Los siguientes requisitos de autenticación y gestión de sesiones mitigan amenazas identificadas en el modelado STRIDE del sistema:

- **REQ-AUTH-01, REQ-AUTH-02, REQ-AUTH-05, REQ-AUTH-07**
  Mitigan **Amenaza #2**: *Un atacante externo o empleado intenta autenticarse o actuar como socio* (Spoofing)

- **REQ-AUTH-03, REQ-AUTH-04**
  Mitigan **Amenaza #6**: *Secuestro de sesión* (Spoofing)

- **REQ-AUTH-01**
  Contribuye a mitigar **Amenaza #1 y #8**: *Repudio interno / negación de acciones* (Repudiation)

- **REQ-AUTH-07**
  Contribuye a reducir la probabilidad de explotación de **Amenaza #6**: *Secuestro de sesión* mediante el uso de credenciales comprometidas.

- **REQ-AUTH-06**
  Reduce el riesgo de explotación de **Amenaza #2** mediante ataques de fuerza bruta o automatizados

3.1.4 Referencias técnicas

- OWASP ASVS 5.0 – V6: Authentication
- OWASP Proactive Controls 2024 – C7: Secure Digital Identities
- CWE-287: Improper Authentication
- CWE-384: Session Fixation
- CWE-620: Unverified Password Change

3.2 Autorización y control de acceso

3.2.1 Objetivo de seguridad

Garantizar que los usuarios del sistema solo puedan acceder a funcionalidades, datos y operaciones estrictamente autorizadas según su rol y contexto, evitando accesos indebidos a información financiera sensible, funciones de administración o acciones que comprometan la integridad del sistema.

El control de acceso deberá implementarse de forma centralizada y validarse siempre del lado servidor, independientemente de las restricciones aplicadas en el frontend.

3.2.2 Requisitos y recomendaciones de diseño

REQ-AUTHZ-01 – Control de acceso basado en roles (RBAC)

El sistema deberá implementar un modelo explícito de control de acceso basado en roles, al menos contemplando los siguientes perfiles:

- Empleado
- Socio
- Administrador

Cada endpoint y operación relevante deberá verificar el rol del usuario autenticado antes de su ejecución.

Antes de la implementación del control de acceso, deberá existir un documento de autorización que defina roles, recursos, permisos y condiciones contextuales relevantes, el cual será la base para las decisiones de autorización en backend.

REQ-AUTHZ-02 – Principio de mínimo privilegio

Cada rol deberá contar únicamente con los permisos estrictamente necesarios para cumplir sus funciones.

No se deberán asignar permisos implícitos o heredados sin justificación funcional explícita.

REQ-AUTHZ-03 – Autorización validada en backend

Las decisiones de autorización deberán realizarse exclusivamente en el backend.

No se deberá confiar en controles de acceso implementados únicamente en el frontend (ocultamiento de botones, rutas, vistas).

REQ-AUTHZ-04 – Separación clara entre roles funcionales y técnicos

Las funciones de negocio (empleado, socio) deberán estar claramente separadas de las funciones técnicas de administración del sistema.

El acceso a funciones administrativas deberá estar restringido a un rol específico y no reutilizar permisos de negocio.

REQ-AUTHZ-05 – Control de acceso a información financiera sensible

El acceso a reportes financieros consolidados, márgenes, caja y datos estratégicos deberá estar restringido exclusivamente a roles autorizados (socios y administrador).

Los empleados no deberán poder acceder directa ni indirectamente a esta información.

REQ-AUTHZ-06 – Autorización contextual para operaciones sensibles

Las operaciones que impacten de forma significativa en la integridad del sistema o la información financiera (por ejemplo: eliminación de gastos, modificación de presupuestos, cambios de estado críticos) deberán:

- Validar el rol del usuario.
- Validar el estado actual del recurso.
- Aplicar controles adicionales cuando corresponda (ver REQ-AUTH-05 y requisitos de autenticación reforzada).

La autorización debe ser dependiente del estado: las transiciones de estado de los recursos financieros deben estar gobernadas por reglas que impidan la edición de datos históricos una vez alcanzado un estado de finalización/cierre. Ejemplo: Si un presupuesto ya fue "Aprobado" o "Facturado", ni siquiera el socio debería poder editar el monto (solo lectura).

REQ-AUTHZ-07 – Prevención de escalamiento horizontal y vertical de privilegios

El sistema deberá prevenir:

- Escalamiento vertical: acceso a funciones de un rol superior.
- Escalamiento horizontal: acceso a recursos pertenecientes a otros usuarios del mismo rol.

Un empleado no debe poder acceder o modificar gastos cargados por otro empleado si no está explícitamente autorizado. El sistema debe validar la propiedad del recurso en cada consulta mediante filtros en la misma query de base de datos, evitando confiar en el ID que viene en la URL.

3.2.3 Trazabilidad con amenazas identificadas

Los siguientes requisitos de autorización y control de acceso mitigan amenazas identificadas en el modelado STRIDE del sistema:

REQ-AUTHZ-01, REQ-AUTHZ-02, REQ-AUTHZ-03, REQ-AUTHZ-04
Mitigan Amenaza #5: *Empleado accede a funciones de socio / Socio accede a funciones de administrador* (Elevation of Privilege)

REQ-AUTHZ-05, REQ-AUTHZ-07
Mitigan Amenaza #4: *Acceso a información sensible por parte de empleados* (Information Disclosure)

REQ-AUTHZ-03, REQ-AUTHZ-07
Mitigan Amenaza #3: *Modificación de datos enviados desde el cliente* (Tampering)

REQ-AUTHZ-06
Contribuye a mitigar Amenaza #7: *Tampering en lógica de negocio*

3.2.4 Referencias técnicas

- OWASP ASVS 5.0 – **V8: Authorization**
- OWASP Proactive Controls 2024 – **C1: Implement Access Control**
- CWE-732: Incorrect Permission Assignment for Critical Resource
- CWE-639: Authorization Bypass Through User-Controlled Key
- CWE-862: Missing Authorization

3.3 Validación de entradas y manejo seguro de datos

3.3.1 Objetivo de seguridad

Garantizar que todos los datos ingresados al sistema —provenientes de usuarios, integraciones o fuentes externas— sean validados, normalizados y procesados de forma segura, evitando modificaciones maliciosas de datos, inyecciones, corrupción de información y alteraciones indebidas de la lógica de negocio.

El sistema no deberá confiar en ningún dato recibido desde el cliente, independientemente de controles aplicados en el frontend.

3.3.2 Requisitos y recomendaciones de diseño

REQ-INPUT-01 – Validación estricta de entradas en backend
Todo dato recibido por el sistema deberá ser validado del lado servidor antes de su procesamiento o persistencia.
La validación deberá contemplar:

- Tipo de dato esperado.

- Longitud máxima y mínima.
- Formato (por ejemplo: fechas, importes, identificadores).
- Rango de valores permitidos.

REQ-INPUT-02 – Enfoque de lista blanca (allowlist)

Las validaciones deberán basarse en valores y formatos explícitamente permitidos. No se deberán utilizar validaciones basadas únicamente en exclusión de caracteres peligrosos (denylist).

REQ-INPUT-03 – Normalización y canonicalización de datos

Los datos deberán ser normalizados antes de su validación y uso, evitando interpretaciones ambiguas del mismo valor (por ejemplo: encoding, espacios, formatos alternativos).

REQ-INPUT-04 – Protección contra inyecciones

El acceso a bases de datos deberá realizarse exclusivamente mediante:

- Consultas parametrizadas.
- ORM o mecanismos equivalentes que separen datos de comandos.
  No se deberá concatenar entrada de usuario en consultas dinámicas.

REQ-INPUT-05 – Manejo seguro de errores y excepciones

Los errores producidos durante la validación o el procesamiento de datos:

- No deberán exponer detalles internos del sistema (queries, stack traces, estructuras internas).
- Deberán registrarse de forma segura para su análisis interno.
- Deberán devolver mensajes genéricos al usuario.

REQ-INPUT-06 – Validación coherente con reglas de negocio

La validación de datos deberá contemplar reglas de negocio relevantes, no solo validaciones sintácticas.
Ejemplos:

- Importes no negativos.
- Fechas coherentes con el período contable.
- Estados válidos según el ciclo de vida del recurso.

REQ-INPUT-07 – Codificación de salida (Output Encoding)

Los datos provenientes de fuentes no confiables deberán ser codificados de forma contextual antes de ser renderizados en el navegador o consumidos por otros sistemas, con el fin de prevenir la ejecución de código no deseado o la interpretación maliciosa de contenido.

3.3.3 Trazabilidad con amenazas identificadas

Los siguientes requisitos de validación y manejo de datos mitigan amenazas identificadas en el modelado STRIDE del sistema:

REQ-INPUT-01, REQ-INPUT-02, REQ-INPUT-03
Mitigan Amenaza #3: Modificación de datos enviados desde el cliente (**Tampering**)

REQ-INPUT-04
Mitiga Amenaza #3 y reduce el riesgo de inyecciones que comprometan integridad y confidencialidad (**Tampering / Information Disclosure**)

REQ-INPUT-05
Contribuye a mitigar Amenaza #4: Exposición de información sensible (**Information Disclosure**)

REQ-INPUT-06
Mitiga Amenaza #7: Manipulación de lógica de negocio mediante datos inconsistentes (**Tampering en lógica de negocio**)

REQ-INPUT-07
Mitiga principalmente:

- Amenaza #4 – Acceso a información sensible **(Information Disclosure**)
  Previene ejecución de scripts inyectados que permitan acceso indebido a datos financieros o información interna.

Contribuye a mitigar:

- Amenaza #6 – Secuestro de sesión (**Spoofing**)
- Amenaza #5 – Elevación de privilegios (**Elevation of Privilege**)

3.3.4 Referencias técnicas

OWASP ASVS 5.0 – **V1: Encoding and Sanitization; V2: Validation and Business Logic**
OWASP Proactive Controls 2024 – **C3: Validate All Input & Handle Exceptions**
CWE-20: Improper Input Validation
CWE-89: SQL Injection
CWE-703: Improper Check or Handling of Exceptional Conditions

3.4 Registro, auditoría y no repudio

3.4.1 Objetivo de seguridad

Garantizar la trazabilidad confiable de las acciones relevantes realizadas en el sistema, permitiendo identificar **quién realizó una acción, sobre qué recurso y en qué momento**, con el fin de reducir el riesgo de repudio interno y facilitar la auditoría, el análisis de incidentes y la investigación de eventos de seguridad.

El sistema deberá registrar de forma consistente las operaciones críticas de autenticación, autorización y gestión de datos financieros, asegurando que los registros generados estén protegidos contra accesos no autorizados y manipulaciones indebidas.

3.4.2 Requisitos y recomendaciones de diseño

REQ-LOG-01 – Registro de eventos críticos

El sistema deberá registrar de forma automática y confiable los siguientes eventos relevantes para la seguridad y la trazabilidad:

- Intentos de autenticación exitosos y fallidos.
- Operaciones de creación, modificación o eliminación de datos financieros (gastos, presupuestos, márgenes, caja).
- Cambios en roles, privilegios de usuario o configuraciones críticas del sistema.
- Acceso y descarga de información financiera sensible

REQ-LOG-02 – Estructura mínima del registro de auditoría

Cada entrada de log deberá contener, como mínimo:

- **Timestamp**: fecha y hora exacta del evento (UTC recomendado).
- **Identificador de usuario**: usuario autenticado que realizó la acción.
- **Tipo de acción**: operación ejecutada (LOGIN, CREATE, UPDATE, DELETE, DOWNLOAD, etc.).
- **Identificador del recurso**: entidad afectada (por ejemplo: Presupuesto ID, Gasto ID).
- **Resultado**: éxito o fallo de la operación.
- **Contexto técnico relevante**: dirección IP de origen u otro identificador de sesión.

REQ-LOG-03 – Inmutabilidad y protección de los logs

Los registros de auditoría deberán:

- Ser de tipo **append-only**, impidiendo su modificación o eliminación desde la aplicación.
- Estar protegidos contra accesos no autorizados.
- No ser accesibles a usuarios finales ni roles de negocio.

REQ-LOG-04 – Acceso restringido y segregación

- El acceso a los registros de auditoría deberá estar restringido exclusivamente al rol Administrador.
- El administrador **no deberá poder modificar ni borrar logs**, únicamente consultarlos.
- Las acciones del propio administrador también deberán quedar registradas.

3.4.3 Trazabilidad con amenazas identificadas

REQ-LOG-01, REQ-LOG-02, REQ-LOG-03, REQ-LOG-04

Mitigan:

- Amenaza #1 y #8 – Repudio interno
- Amenaza #13 – Borrado o alteración de logs **(Tampering)**

Además contribuyen a la detección temprana de:

- Amenaza #3 (**Tampering**)
- Amenaza #5 (**Elevation of Privilege)**

3.4.4 Referencias técnicas

- OWASP ASVS 5.0 – **V16 Security Logging and Error Handling**
- OWASP Proactive Controls 2024 – **C9: Implement Security Logging and Monitoring**
- CWE-778: Insufficient Logging
- CWE-117: Improper Output Neutralization for Logs

3.5 Protección de datos sensibles y criptografía

3.5.1 Objetivo de seguridad

Garantizar la confidencialidad e integridad de los datos sensibles gestionados por el sistema —en particular información financiera, datos de clientes y credenciales— tanto en reposo como en tránsito, evitando accesos no autorizados, exposiciones accidentales o manipulaciones indebidas.

El sistema deberá aplicar mecanismos criptográficos adecuados y proporcionales al riesgo para proteger los datos críticos, asegurando que su uso, almacenamiento y transmisión se realicen de forma segura y consistente, sin depender de controles del lado cliente ni de supuestos implícitos de confianza.

3.5.2 Requisitos y recomendaciones de diseño

REQ-DATA-01 – Gestión segura de secretos y llaves

No se deberán incluir credenciales, secretos criptográficos ni llaves de acceso (por ejemplo: credenciales de base de datos, API keys, tokens, secretos de firma) directamente en el código fuente ni en repositorios de control de versiones.

Los secretos deberán gestionarse mediante mecanismos seguros, tales como:

- Variables de entorno protegidas.
- Servicios de gestión de secretos o vaults, cuando estén disponibles.

El acceso a los secretos deberá limitarse únicamente a los componentes que lo requieran para su funcionamiento.

REQ-DATA-02 – Protección de documentos y archivos sensibles

Los documentos generados por el sistema que contengan información sensible (por ejemplo: presupuestos, contratos, reportes financieros en PDF) no deberán almacenarse en ubicaciones con acceso público directo.

El acceso a dichos archivos deberá:

- Estar mediado por un controlador de la aplicación.
- Validar la sesión activa del usuario.
- Verificar los permisos y la autorización sobre el recurso solicitado (propiedad, rol y contexto).

No se deberán utilizar URLs predecibles ni rutas directas al sistema de archivos para servir documentos sensibles.

REQ-DATA-03 – Comunicaciones seguras

Toda la comunicación entre clientes y el servidor deberá realizarse exclusivamente mediante protocolos cifrados. En particular:

- Se deberá utilizar TLS versión 1.2 o superior.
- No se deberán permitir conexiones mediante HTTP sin cifrado.

El sistema deberá implementar mecanismos para reforzar el uso de canales seguros, tales como HTTP Strict Transport Security (HSTS) cuando aplique.

REQ-DATA-04 – Uso adecuado de criptografía estándar

Cuando se utilicen mecanismos criptográficos para proteger datos sensibles, se deberán emplear:

- Algoritmos y bibliotecas criptográficas estándar, ampliamente revisadas y mantenidas.
- Configuraciones seguras por defecto, evitando parámetros obsoletos o débiles.

No se deberán implementar algoritmos criptográficos propios ni mecanismos de cifrado “caseros”.

3.5.3 Trazabilidad con amenazas identificadas

Los siguientes requisitos de protección de datos y criptografía mitigan amenazas identificadas en el modelado STRIDE del sistema:

REQ-DATA-01

Mitiga:

- Amenaza #10 – Acceso no autorizado a tablas sensibles (**Information Disclosure**)
- Amenaza #2 – Uso indebido de credenciales para suplantación (**Spoofing**)

REQ-DATA-02

Mitiga:

- Amenaza #12 – Acceso a PDFs sin autorización (**Information Disclosure**)

- Amenaza #4 – Acceso a información financiera sensible por empleados (**Information Disclosure**)

REQ-DATA-03
Mitiga:

- Amenaza #6 – Secuestro de sesión mediante interceptación de comunicaciones (**Spoofing**)
- Amenaza #4 – Exposición de información sensible en tránsito (**Information Disclosure**)

REQ-DATA-04
Reduce el riesgo transversal de:

- Exposición de información sensible por uso incorrecto de criptografía (Information Disclosure)
- Compromiso de integridad de datos protegidos (Tampering)

3.5.4 Referencias técnicas

- OWASP ASVS 5.0 – **V11: Cryptography; V9: Secure communication**
- OWASP Proactive Controls 2024 – **C2: Leverage Security Frameworks and Libraries; C4: Secure Data Storage**
- CWE-321: Use of Hard-coded Cryptographic Key
- CWE-522: Insufficiently Protected Credentials
- CWE-319: Cleartext Transmission of Sensitive Information