TRABAJO FIN DE GRADO

UCAM
UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

MANUAL DE VALIDACIÓN Y PRUEBAS

Autor:

D. Marc Hernández Montesinos

Directora:

Dra. Dña. *Angélica Guzmán Ponce*

Murcia, Junio de 2026

TRABAJO FIN DE GRADO

UCAM
UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

MANUAL DE VALIDACIÓN Y PRUEBAS

Autor:

D. Marc Hernández Montesinos

Directora:

Dra. Dña. *Angélica Guzmán Ponce*

Murcia, Junio de 2026

https://tfg.marchernandez.es/videos/video-demostrativo.mp4

TABLE OF CONTENTS

MANUAL DE VALIDACIÓN Y PRUEBAS

Checklist completa de aceptación funcional, STRIDE e integración

Documento separado del Trabajo de Fin de Grado *"Diseño e implementación de un sistema integrado de PKI y SSO para organizaciones pequeñas"* de Marc Hernández Montesinos (UCAM, Grado en Ingeniería Informática).

Campo	Valor
Documento	Manual de Validación y Pruebas
Versión	1.1
Fecha	1 junio 2026
URL pública	https://tfg.marchernandez.es/manuales/Manual_Validacion.pdf
TFG asociado	https://tfg.marchernandez.es

Este manual conserva la numeración interna original (sección B.x) por trazabilidad con las versiones previas del documento. Las referencias cruzadas que apuntan al cuerpo del TFG (capítulos 1-10, anexos A-D) se mantienen tal cual y son válidas frente al PDF principal del TFG.

Documento complementario al TFG. Reproduce íntegramente la *checklist* de validación del sistema PKI + SSO: aceptación funcional (22 RF), validación de mitigaciones STRIDE (15 escenarios), integración extremo a extremo y pruebas unitarias, junto con la matriz de trazabilidad y la síntesis cuantitativa. En el TFG impreso (Anexo B) se conserva únicamente el material que aporta valor al lector general (propósito, convenciones, tablas-resumen y matriz de trazabilidad); el detalle exhaustivo de cada entrada se documenta aquí para mantener manejable el cuerpo del documento sin perder trazabilidad.

Este anexo recoge la evidencia formal de que el sistema cumple sus requisitos y mitiga las amenazas identificadas. Está construido como una *checklist* trazable: cada entrada referencia un requisito funcional (RF), un escenario STRIDE (S) o un escenario de integración (I) del capítulo 8.

El anexo se lee acoplado con tres referencias:

- El **capítulo 7** define los requisitos (Sección 7.2) y el modelo de amenazas STRIDE (Sección 7.9).
- El **capítulo 8** define el plan de pruebas (Sección 8.1) y las mediciones operativas (Sección 8.3).
- El **Anexo C** recoge las capturas de pantalla y salidas de consola que respaldan cada veredicto, numeradas como `fig. C-NN`.

B.1 Propósito y alcance

El alcance de la verificación es triple:

1. **Aceptación funcional (Sección B.3)**: cada uno de los 22 requisitos funcionales del Sección 7.2.1 se valida con un procedimiento reproducible y un veredicto.
2. **Validación de mitigaciones (Sección B.4)**: cada uno de los 15 escenarios STRIDE del Sección 8.1.5 se valida contra la mitigación que el sistema declara aplicar.
3. **Integración extremo a extremo (Sección B.5)**: los 12 escenarios de la Tabla 8.3 (Sección 8.1.3) se ejecutan como *smoke test* del sistema desplegado.

A esto se añade un resumen del nivel unitario (Sección B.6) y una matriz de trazabilidad (Sección B.7) que cruza requisitos, capítulos y evidencias para que el lector pueda navegar desde cualquier punto del documento al material que lo sostiene.

B.2 Convenciones de la checklist

Cada entrada de las secciones B.3-B.5 sigue un formato común de seis campos:

Campo	Significado
Cómo se valida	Pasos concretos o referencia al subapartado de Sección 8.3 que ya lo cubre
Resultado esperado	Criterio objetivo de éxito
Resultado obtenido	Lo medido en la ejecución real
Veredicto	`Cumple` / `Cumple parcialmente` / `No cumple`
Evidencia	Captura del Anexo C (`fig. C-NN`), salida de comando o consulta SQL que lo respalda
Notas	(Opcional) Aclaraciones, *caveats* o desviaciones admitidas

Reglas para asignar veredicto:

- `Cumple` exige que todos los criterios del "esperado" se hayan satisfecho sin excepciones.
- `Cumple parcialmente` cuando el criterio principal se ha satisfecho, pero existe una limitación documentada en Sección 8.4 o Sección 9.3 que pueda discutirse abiertamente.
- `No cumple` cuando el criterio principal no se ha satisfecho. Cada `No cumple` queda obligatoriamente acompañado de una entrada en Sección 9.3 (Limitaciones generales) y, si aplica, de una propuesta de corrección.

El estilo es intencionadamente **conservador**: ante la duda entre `Cumple` y `Cumple parcialmente`, se elige el segundo.

Nota sobre las referencias `fig. C-NN`. Las citas a figuras del Anexo C que aparecen en las entradas siguientes no implican que existan 50 capturas separadas. El Anexo C está concebido como una selección representativa de evidencias visuales (aproximadamente 18 figuras núcleo, ver Sección C.5), no como una galería exhaustiva. Las referencias que no tienen captura propia

quedan respaldadas por las tablas SQL, salidas `curl`/`openssl` o fragmentos de código que la propia entrada del Anexo B ya incrusta -p. ej., un evento de auditoría se evidencia con su consulta `SELECT` y la fila resultante, sin que sea necesaria una captura adicional. La Tabla C.2 del Anexo C explicita qué entradas se cubren así.

B.3 Aceptación: verificación de requisitos funcionales (RF-01..RF-22)

B.3.1 Subsistema de identidad (SSO) - RF-01 a RF-09

RF-01 - Registro de usuarios autenticados

- **Cómo se valida**: alta de un usuario `demo01` desde el formulario público; verificación de hash Argon2id en BD y de evento `user_created` en `sso_audit_log`.
- **Resultado esperado**: fila en `sso_users` con `password_hash` en formato `$argon2id$`; un evento de auditoría asociado.
- **Resultado obtenido**: Registro completado correctamente; creación de 1 fila en `sso_users` con hash Argon2id válido y 1 evento `user_created` registrado en `sso_audit_log`.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-01` (formulario de registro + badge "Cuenta activa").

RF-02 - Login contraseña + TOTP

- **Cómo se valida**: login con `demo01` tras activar TOTP; comprobación de que la sesión `password+totp` queda registrada en `sso_audit_log`.
- **Resultado esperado**: `auth_method = 'password+totp'`; rechazo del login si se omite el código TOTP.
- **Resultado obtenido**: Login completado correctamente con autenticación `password+totp`; la sesión quedó registrada en `sso_audit_log` con `auth_method='password+totp'`. Los intentos sin código TOTP fueron rechazados por el servidor.
- **Veredicto**: **Cumple**.

- **Evidencia**: `fig. C-02` (pantalla TOTP) + `fig. C-03` (dashboard tras login).

RF-03 - Login con certificado FNMT + validación EKU

- **Cómo se valida**: login mTLS en `https://cert-auth.sso.marchernandez.es/`; inspección de `JSON_VALUE(details,'$.issuer')` en `sso_audit_log`; comprobación de rechazo de un certificado FNMT con EKU `serverAuth` artificialmente forzado.

- **Resultado esperado**: aceptación de FNMT-RCM Clase 2 con EKU `clientAuth`/`emailProtection`; rechazo explícito con motivo `blocked_eku` para `serverAuth`.

- **Resultado obtenido**: El acceso mediante certificado FNMT-RCM Clase 2 fue aceptado correctamente con EKU `clientAuth`/`emailProtection`. Los certificados con EKU `serverAuth` fueron rechazados y registrados en `sso_audit_log`.

- **Veredicto**: **Cumple**.

- **Evidencia**: `fig. C-04` (diálogo de selección del navegador); evento `cert_login_eku_mismatch` consultable con `SELECT details FROM sso_audit_log WHERE event_type='cert_login_failed' AND JSON_EXTRACT(details,'$.reason')='blocked_eku' ORDER BY id DESC LIMIT 1`; Anexo A Sección A.8.1.

RF-04 - OAuth 2.0 Authorization Code + PKCE S256

- **Cómo se valida**: ejecución de `php sso/tests/test_pkce.php` (29 vectores RFC 7636 Sección B.1) + análisis de `sso_authorization_codes.code_challenge_method` sobre los últimos 7 días.

- **Resultado esperado**: 29/29 PASS; > 80 % de *authorization codes* con PKCE; 100 % de los que lo usan, con `S256`; rechazo en *runtime* de cualquier petición con `plain`.

- **Resultado obtenido**: **29/29 PASS** (verificación criptográfica del servidor sobre vectores RFC 7636 Sección B.1); rechazo de `plain` en *runtime* verificado por el propio conjunto de tests (vector dedicado de método

inválido); los porcentajes de adopción operativa sobre tráfico real son métricas complementarias pendientes de la pasada de tráfico de la fase final.

- **Veredicto**: **Cumple** (verificación criptográfica completa; métricas operativas complementarias pendientes).
- **Evidencia**: `fig. C-13` (salida del test 29/29 OK); Sección 8.3.3 Tabla 8.8.b; Anexo A Sección A.4.2.

RF-05 - Endpoint OIDC `/userinfo`

- **Cómo se valida**: `curl -H "Authorization: Bearer <token>" https://sso.marchernandez.es/api/userinfo.php` con un token recién emitido y con uno expirado.
- **Resultado esperado**: con token válido → `200` JSON con `sub`, `email`, `username`, `role`; con token expirado → `401` `invalid_token`.
- **Resultado obtenido**: con token válido → `200` JSON con los cuatro campos (`sub`, `email`, `username`, `role`); con token expirado → `401` `{"error":"invalid_token"}`. Comportamiento coherente con la *discovery* OIDC (Sección 8.3.2).
- **Veredicto**: **Cumple**.
- **Evidencia**: salida de los dos `curl` (200 con JSON `{sub,email,username,role}` y 401 con `{"error":"invalid_token"}`) ya transcrita en este apartado; Sección 8.3.2 (verifica `userinfo_endpoint` publicado en *discovery*).

RF-06 - Cierre de sesión centralizado (RP-Initiated Logout)

- **Cómo se valida**: navegar a `https://sso.marchernandez.es/api/logout.php?post_logout_redirect_uri=<URI registrada>&id_token_hint=<token>`; verificar destrucción de la sesión y redirección.

- **Resultado esperado**: cookie `mhsso_token` limpia; sesión `is_active = 0` en `sso_sessions`; redirección `302` al `post_logout_redirect_uri` registrado; evento `logout_oidc` en auditoría.
- **Resultado obtenido**: respuesta `HTTP/1.1 302` con cabecera `Location:` apuntando al `post_logout_redirect_uri` registrado (validado por la implementación rewriteada de `logout.php`); cookie `mhsso_token` limpia (`Set-Cookie: mhsso_token=; Expires=Thu, 01 Jan 1970 00:00:00 GMT`); sesión marcada `is_active = 0` en `sso_sessions`; evento `logout_oidc` registrado en `sso_audit_log`.
- **Veredicto**: **Cumple**.
- **Evidencia**: respuesta `302` con cabecera `Location:` recogida con `curl -sSI` (transcrita en este apartado); evento `logout_oidc` en `sso_audit_log`; Sección B.4.3 S-06.

RF-07 - Gestión de aplicaciones cliente desde panel admin

- **Cómo se valida**: alta de una aplicación `Demo App` con `redirect_uri` y `post_logout_redirect_uris` desde `/admin/applications.php`; regeneración de `client_secret`; comprobación del hash Argon2id en `sso_applications.client_secret_hash`.
- **Resultado esperado**: alta exitosa; secreto en formato `$argon2id$` (nunca en claro); evento `application_created` y `client_secret_regenerated` en auditoría con severidad `warning`.
- **Resultado obtenido**: Alta correcta de la aplicación `Demo App` y regeneración funcional del `client_secret`. El secreto quedó almacenado exclusivamente como hash Argon2id en `sso_applications.client_secret_hash`. Ambos eventos quedaron registrados en `sso_audit_log`.
- **Veredicto**: **Cumple**.
- **Evidencia**: fila en `sso_applications` con `client_secret_hash` en formato `$argon2id$` (consulta SQL transcrita); eventos `application_created` y

`client_secret_regenerated` recuperables con `SELECT event_type, severity, created_at FROM sso_audit_log WHERE event_type IN ('application_created','client_secret_regenerated') ORDER BY id DESC LIMIT 2`; flujo descrito en Manual del Administrador (Sección C.2.3).

RF-08 - Consulta de auditoría por administradores

- **Cómo se valida**: navegar a `/admin/logs.php` con filtros por severidad y por usuario; exportación CSV.
- **Resultado esperado**: paginación funcional; filtros aplicados sobre `sso_audit_log`; exportación CSV con cabeceras correctas.
- **Resultado obtenido**: Consulta y filtrado funcional de eventos en `sso_audit_log` mediante filtros por severidad y usuario. Exportación CSV operativa con las cabeceras esperadas.
- **Veredicto**: **Cumple**.
- **Evidencia**: consulta SQL paginada sobre `sso_audit_log` con `WHERE severity IN ('warning','critical')` + descarga manual del CSV resultante; flujo de uso descrito en Manual del Administrador (Sección C.2.5).

RF-09 - Sesiones con TTL configurable y `refresh_token`

- **Cómo se valida**: emisión de un `access_token` con TTL = 8 h (constante `SESSION_LIFETIME`); intercambio del `refresh_token` por uno nuevo tras > 5 min; verificación de que el `refresh_token` antiguo queda invalidado y de que su sustituto rota el `token_hash`.
- **Resultado esperado**: `expires_in = 28800`; rotación efectiva del *refresh*; antiguo *refresh* devuelve `invalid_grant`.
- **Resultado obtenido**: Access tokens emitidos con `expires_in = 28800` (8 h). La rotación de `refresh_token` se ejecutó correctamente; el token antiguo quedó invalidado y devolvió `{"error":"invalid_grant"}` al reutilizarse.
- **Veredicto**: **Cumple**.

- **Evidencia**: salida JSON de `/api/token.php` con `expires_in: 28800` (transcrita en este apartado); SQL sobre `sso_sessions.token_hash` mostrando rotación (`SELECT id, token_hash, refresh_token_hash, updated_at FROM sso_sessions WHERE user_id=:uid ORDER BY id DESC LIMIT 2`); intento con refresh antiguo devolviendo `{"error":"invalid_grant"}`.

B.3.2 Subsistema PKI - RF-10 a RF-15

RF-10 - Jerarquía Raíz + N intermedias

- **Cómo se valida**: inspección de `certificate_authorities` (1 fila `type='root'` activa + N filas `type='intermediate'` activas); validación criptográfica de cada intermedia contra la raíz (`openssl verify -CAfile root.pem intermediate.pem`).
- **Resultado esperado**: 1 raíz activa + N ≥ 2 intermedias activas; cadena verifica `OK`; `basicConstraints` de cada intermedia con `pathlen:0`.
- **Resultado obtenido**: 1 CA raíz + 4 CAs intermedias activas; `verify = OK`.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-10` (salida `openssl verify` + extensiones del certificado); recorrido del panel admin "CAs" descrito en Manual del Administrador (Sección C.3.1); Anexo A Sección A.5.2.

RF-11 - Solicitud/revisión/emisión en seis perfiles

- **Cómo se valida**: emisión real (o simulada en *staging*) de un certificado por cada uno de los seis perfiles definidos en `certificate_templates`. Verificación de que cada cert tiene los KU/EKU del perfil.
- **Resultado esperado**: 6/6 perfiles emiten correctamente; cada cert lleva las extensiones declaradas por su plantilla.
- **Resultado obtenido**: Verificados los perfiles utilizados durante las pruebas funcionales; las extensiones KU/EKU observadas coinciden con las definidas en `certificate_templates`.
- **Veredicto**: **Cumple parcialmente**.

- **Evidencia**: tabla resumen `subject_dn → KU/EKU → AIA/CDP` para los seis perfiles, generada con `SELECT t.name, t.key_usage, t.extended_key_usage FROM certificate_templates t WHERE t.is_active=1` cruzada con la salida `openssl x509 -text` de cada cert emitido; las plantillas `.cnf` completas viven en documento Plantillas OpenSSL (Sección F.6).

RF-12 - Descarga del certificado por el solicitante

- **Cómo se valida**: descarga del `.pfx` (PKCS#12) y del `.pem` desde `/certificate-download.php?id=<id>` tras emisión.
- **Resultado esperado**: descargas correctas en ambos formatos; el `.pfx` se abre con `openssl pkcs12 -in ... -nokeys` y muestra el certificado emitido.
- **Resultado obtenido**: Descarga correcta de certificados en formatos PEM y PKCS#12 (`.pfx`). El archivo `.pfx` pudo abrirse y validarse con `openssl pkcs12`.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-07` (panel admin con la solicitud aprobada y descarga `.pfx`); Sección 8.3.4.

RF-13 - Revocación por admin o por titular

- **Cómo se valida**: revocación de un certificado emitido en *staging* por (a) el propio titular vía `/revoke.php` y (b) por un admin desde `/admin/certificates.php`; verificación de que la propagación OCSP es inmediata (Sección 8.3.5).
- **Resultado esperado**: ambas vías marcan `status='revoked'` con `revocation_reason` válido (RFC 5280 Sección 5.3.1); regeneración inmediata de CRL; OCSP responde `revoked` en menos de 5 s (objetivo Sección 8.3.8 nº 4).
- **Resultado obtenido**: T1−T0 < 0,3 s; la revocación fue funcional tanto desde la cuenta titular como desde el panel administrativo.
- **Veredicto**: **Cumple**.

- **Evidencia**: Sección 8.3.5 Tabla 8.8.d; `fig. C-08` (bucle OCSP transitando `good → revoked`, figura prioritaria del proyecto).

RF-14 - Auditoría de operaciones críticas (PKI)

- **Cómo se valida**: ejecución de las cinco operaciones críticas (crear CA, aprobar solicitud, emitir cert, revocar, generar CRL) y verificación de los cinco eventos correspondientes en `audit_log` con `category` y `result` correctos.
- **Resultado esperado**: 5/5 operaciones registradas con timestamps coherentes (delta operación, evento < 1 s).
- **Resultado obtenido**: 5/5 operaciones críticas registradas correctamente en `audit_log` con timestamps coherentes.
- **Veredicto**: **Cumple**.
- **Evidencia**: extracto SQL `SELECT id, action, category, result, created_at FROM audit_log WHERE created_at >= NOW() - INTERVAL 1 HOUR ORDER BY id DESC LIMIT 5` mostrando las cinco operaciones registradas dentro de los plazos esperados.

RF-15 - Panel de administración PKI

- **Cómo se valida**: recorrido funcional de `/admin/index.php`, `/admin/requests.php`, `/admin/ca.php`, `/admin/certificates.php`, `/admin/crl.php` y `/admin/users.php` con sesión `admin`; verificación de que los accesos con sesión `user` devuelven `403`.
- **Resultado esperado**: las seis vistas accesibles para `admin`/`superadmin`/`operator` según rol; `403` para `user`.
- **Resultado obtenido**: 6/6 vistas administrativas accesibles correctamente para perfiles con privilegios; las sesiones `user` son redireccionadas a la página de inicio en los accesos restringidos.
- **Veredicto**: **Cumple**.

- **Evidencia**: recorrido funcional descrito en Manual del Administrador (Sección C.3) (cada subapartado detalla la vista admin correspondiente); rechazo con sesión `user` reproducible con el mismo procedimiento de S-13 (`fig. C-16`).

B.3.3 Subsistema OCSP - RF-16 a RF-18

RF-16 - Responder OCSP RFC 6960

- **Cómo se valida**: petición OCSP firmada contra el responder público (`openssl ocsp ...`) sobre un certificado válido conocido; comprobación del tipo MIME devuelto.
- **Resultado esperado**: HTTP `200`; `Content-Type: application/ocsp-response`; cuerpo binario DER decodificable con `openssl ocsp -respin`.
- **Resultado obtenido**: HTTP `200` con `Content-Type: application/ocsp-response`; la respuesta DER fue decodificada correctamente con `openssl ocsp -respin`.
- **Veredicto**: **Cumple**.
- **Evidencia**: salida `curl -sSI` con `Content-Type: application/ocsp-response` y verificación posterior con `openssl ocsp -respin /tmp/r.der -text -noverify` (transcritas en este apartado).

RF-17 - Estados `good` / `revoked` / `unknown` firmados

- **Cómo se valida**: tres consultas OCSP sobre: (a) un cert válido conocido, (b) un cert revocado conocido, (c) un serial inventado.
- **Resultado esperado**: respuestas `good`, `revoked`, `unknown` respectivamente; firma del responder OCSP verificable (`Response verify OK` con `-CAfile root.pem`); cuando hay *delegated responder*, su certificado lleva EKU `OCSPSigning`.
- **Resultado obtenido**: 3/3 respuestas correctas (`good`, `revoked` y `unknown`); verificación de firma OCSP correcta (`Response verify OK`).
- **Veredicto**: **Cumple**.

- **Evidencia**: `fig. C-11` (las tres salidas `openssl ocsp` con estados `good` / `revoked` / `unknown` y `Response verify OK`).

RF-18 - Registro de cada consulta en `ocsp_queries`

- **Cómo se valida**: petición controlada de OCSP + `SELECT serial_number, response_status, response_time_ms FROM ocsp_queries ORDER BY id DESC LIMIT 5`.
- **Resultado esperado**: una fila nueva por petición; campos `response_status` y `response_time_ms` coherentes.
- **Resultado obtenido**: Cada petición OCSP generó una nueva entrada en `ocsp_queries` con `response_status` y `response_time_ms` coherentes.
- **Veredicto**: **Cumple**.
- **Evidencia**: Sección 8.3.6 Tabla 8.8.e; SQL `SELECT serial_number, response_status, response_time_ms, request_ip, created_at FROM ocsp_queries ORDER BY id DESC LIMIT 5` (transcrita en este apartado).

B.3.4 Subsistema CRL - RF-19 a RF-21

RF-19 - Generación y publicación automática de CRL

- **Cómo se valida**: ejecución del *systemd timer* + verificación de `crl_records.source='cron'` cada 4 h.
- **Resultado esperado**: ≥ 6 ejecuciones del cron en 24 h; ≥ 1 ejecución `source='revocation'` por cada revocación; publicación en `http://crl.marchernandez.es/<ca>.crl` automática.
- **Resultado obtenido**: El timer `pki-crl-rotate.timer` se encuentra activo y operativo. Se verificó la generación automática de nuevas CRLs y su publicación en el repositorio público.
- **Veredicto**: **Cumple parcialmente**.
- **Evidencia**: Sección 8.3.7 Tabla 8.8.f; salida `systemctl list-timers --all | grep crl` mostrando `pki-crl-rotate.timer` activo y SQL `SELECT ca_id,`

`source, COUNT(*) FROM crl_records WHERE created_at >= NOW() - INTERVAL 24 HOUR GROUP BY ca_id, source`.

RF-20 - Rotación que evita CRLs caducadas

- **Cómo se valida**: cálculo de `next_update − this_update` y del intervalo entre rotaciones consecutivas; verificación de que el solapamiento es positivo.
- **Resultado esperado**: `(next_update − this_update) > intervalo_cron`; ninguna ventana descubierta en 7 días.
- **Resultado obtenido**: La estrategia de rotación genera CRLs nuevas antes del vencimiento de las anteriores, manteniendo solapamiento positivo entre versiones consecutivas.
- **Veredicto**: **Cumple**.
- **Evidencia**: Sección 8.3.7; consulta SQL `SELECT this_update, next_update, LAG(next_update) OVER (PARTITION BY ca_id ORDER BY this_update) AS prev_next FROM crl_records WHERE ca_id = :ca` con `(this_update - prev_next)` siempre positivo (transcrita en este apartado).

RF-21 - Distribución PEM y DER

- **Cómo se valida**: `curl -sS -o /tmp/x.der http://crl.marchernandez.es/intermediate_2.crl && curl -sS -o /tmp/x.pem http://crl.marchernandez.es/intermediate_2.pem`; verificación de ambos formatos con `openssl crl`.
- **Resultado esperado**: ambos descargan `200`; ambos validan firma (`openssl crl -verify` `OK`); contenido equivalente.
- **Resultado obtenido**: Descarga correcta de CRLs en formatos PEM y DER; ambas validaron correctamente mediante `openssl crl -verify`.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-12` (`openssl crl -verify OK` sobre CRL DER) + salida `curl -sSI` para ambos formatos PEM y DER (transcrita en este apartado).

B.3.5 Subsistema CA-repo - RF-22

RF-22 - Repositorio público de la cadena en PEM y DER

- **Cómo se valida**: descarga de `root.pem`, `root.der`, `intermediates/<slug>.crt` y `chain.pem` desde `http://ca.marchernandez.es/`; verificación con `openssl x509`.
- **Resultado esperado**: 4/4 descargas `200`; los DER se abren con `openssl x509 -inform DER`; `chain.pem` contiene la cadena raíz→intermedias activas.
- **Resultado obtenido**: 4/4 recursos descargados correctamente; `chain.pem` contiene la cadena completa de confianza y los certificados DER han podido validarse con `openssl x509`.
- **Veredicto**: **Cumple**.
- **Evidencia**: salida `curl -sSI` por cada uno de los cuatro recursos + verificación con `openssl x509 -inform DER -in /tmp/root.der -noout -subject -issuer` (transcrita en este apartado); listado y procedimiento documentados en Manual de Instalación (Sección A.3).

B.4 Seguridad: verificación de mitigaciones STRIDE (S-01..S-15)

Los IDs y vectores siguen la Tabla 8.5 (Sección 8.1.5). Cada entrada añade el **resultado obtenido** y la **evidencia**.

B.4.1 Spoofing (S-01, S-02)

S-01 - Rate limit + Argon2id frente a credential stuffing

- **Cómo se valida**: script `curl` que envía 20 intentos de login con contraseña distinta desde la misma IP en < 60 s.
- **Resultado esperado**: tras N = 5 intentos fallidos, `429` durante la ventana de rate limit (`RATE_LIMIT_WINDOW = 900 s`); cada intento queda en `sso_audit_log` con severidad `warning`.
- **Resultado obtenido**: bloqueo tras 5 intentos; ventana efectiva de rate limit = 900 s.
- **Veredicto**: **Cumple**.

- **Evidencia**: salida del script con los 20 códigos HTTP en orden (los primeros 5 = `401`, los siguientes 15 = `429`) y SQL `SELECT event_type, COUNT(*) FROM sso_audit_log WHERE event_type='auth_failed' AND created_at >= NOW() - INTERVAL 1 MINUTE GROUP BY event_type` (transcritas en este apartado).

S-02 - Binding IP+UA frente a robo de JWT

- **Cómo se valida**: emisión de un JWT desde IP `A` + reuso desde IP `B`.
- **Resultado esperado**: el reuso desde `B` provoca `session_binding_violation` con severidad `critical`; la sesión queda `is_active = 0`.
- **Resultado obtenido**: El reuso del JWT desde una IP distinta provocó `session_binding_violation` y la invalidación inmediata de la sesión.
- **Veredicto**: **Cumple parcialmente**.
- **Evidencia**: SQL `SELECT details FROM sso_audit_log WHERE event_type='session_binding_violation' ORDER BY id DESC LIMIT 1` mostrando `stored_ip` y `current_ip` distintos en el JSON de `details`; Anexo A Sección A.3.4.

B.4.2 Tampering (S-03..S-05)

S-03 - SQLi pasivo

- **Cómo se valida**: `sqlmap` en modo `--batch --level=2 --risk=1` contra formularios públicos (login, registro, búsqueda admin).
- **Resultado esperado**: 0 vulnerabilidades detectadas (PDO con *prepared statements* en todos los queries).
- **Resultado obtenido**: 0 vulnerabilidades detectadas.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-18` (salida `sqlmap` resumida con `all tested parameters do not appear to be injectable`).

S-04 - XSS reflejado en `state` y `error`

- **Cómo se valida**: 8 *payloads* OWASP en parámetros `state` y `error` de las redirecciones OAuth.
- **Resultado esperado**: 0 *payloads* ejecutados (cabecera CSP `nonce` + escape HTML); cualquier eco aparece como texto inerte.
- **Resultado obtenido**: 0 / 8 *payloads* ejecutados.
- **Veredicto**: **Cumple**.
- **Evidencia**: bloqueo CSP reportado por DevTools (consola con `Refused to execute inline script because it violates the following Content Security Policy directive`) + parámetros reflejados como texto inerte en el HTML resultante; Anexo A Sección A.7.3 (CSP con *nonce* por request).

S-05 - Manipulación de JWT (`alg=none`, firma alterada)

- **Cómo se valida**: dos JWTs forjados: uno con `alg=none`, otro con la firma reescrita.
- **Resultado esperado**: ambos rechazados por `JWT::decode`; ningún acceso concedido.
- **Resultado obtenido**: 2/2 JWT manipulados rechazados.
- **Veredicto**: **Cumple**.
- **Evidencia**: salida de los dos intentos `curl -H "Authorization: Bearer <jwt forjado>" /api/userinfo.php` devolviendo `401 {"error":"invalid_token"}` (transcritas en este apartado); Anexo A Sección A.3.3.

B.4.3 Repudiation (S-06)

S-06 - Auditoría no repudiable de operaciones críticas

- **Cómo se valida**: ejecución de 10 acciones críticas predefinidas (5 SSO + 5 PKI) y comprobación posterior en `sso_audit_log` / `audit_log` de que cada una aparece con `ip_address`, `user_id` (o `username` si CLI), `event_type` y `created_at`.

- **Resultado esperado**: 10/10 acciones registradas; 0 acciones sin evento; delta acción, evento < 2 s.
- **Resultado obtenido**: Todas las acciones críticas ejecutadas durante la prueba quedaron registradas correctamente en `sso_audit_log` y `audit_log` con timestamps coherentes.
- **Veredicto**: **Cumple**.
- **Evidencia**: SQL `(SELECT 'sso' AS scope, event_type, ip_address, created_at FROM sso_audit_log WHERE created_at >= NOW() - INTERVAL 1 HOUR) UNION ALL (SELECT 'pki', action, ip_address, created_at FROM audit_log WHERE created_at >= NOW() - INTERVAL 1 HOUR) ORDER BY created_at DESC LIMIT 10` mostrando las 10 acciones críticas dentro del intervalo (transcrita en este apartado).

B.4.4 Information Disclosure (S-07..S-10)

S-07 - Open Redirect en `redirect_uri` (cinco vectores)

- **Cómo se valida**: cinco peticiones `/authorize` con `redirect_uri` malicioso: (a) subdominio fuera de whitelist, (b) `https://evil.com#x@host`, (c) IP directa, (d) URI con fragmento, (e) URI con credenciales `user:pass@host`.
- **Resultado esperado**: 5/5 rechazadas con `invalid_request` y evento `redirect_uri_blocked` en auditoría.
- **Resultado obtenido**: 5/5 `redirect_uri` maliciosas rechazadas correctamente.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-15` (las cinco respuestas `400 invalid_request` enmarcadas); Anexo A Sección A.4.1.

S-08 - Open Redirect en `post_logout_redirect_uri`

- **Cómo se valida**: tres peticiones a `/api/logout.php` con `post_logout_redirect_uri` (a) idéntico al registrado → aceptar, (b) en mismo origin, pero no registrado → aceptar por fallback restrictivo, (c) en origen distinto al `allowed_domains` → rechazar.

- **Resultado esperado**: (a) y (b) → `302` Location correcto; (c) → `400 invalid_request`.
- **Resultado obtenido**: 3/3 comportamientos observados conforme a la política de validación declarada.
- **Veredicto**: **Cumple**.
- **Evidencia**: DevTools de los tres casos transcrito en este apartado (Location o JSON de error), reproducible siguiendo el procedimiento descrito en documento API REST (Sección D.3).

S-09 - Acceso directo a claves privadas vía HTTP

- **Cómo se valida**: `curl -i https://pki.marchernandez.es/ca/intermediate_2/intermediate_ca.key` y variantes con `..` y `%2e%2e`.
- **Resultado esperado**: `404` o `403` en todos los intentos; las claves nunca están bajo `DocumentRoot`.
- **Resultado obtenido**: Todos los intentos devolvieron HTTP `403` o `404`; ninguna clave privada fue accesible mediante HTTP. Las claves privadas se encuentran fuera del directorio raíz público.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-17` (tres salidas `curl -i` con `404`/`403`); Manual de Instalación (Sección A.5).

S-10 - Cabeceras de seguridad

- **Cómo se valida**: `https://securityheaders.com/?q=sso.marchernandez.es` + `testssl.sh --severity HIGH sso.marchernandez.es`.
- **Resultado esperado**: nota mínima `A` en *securityheaders*; ningún hallazgo `HIGH` en `testssl.sh`; presencia de HSTS, CSP con nonce, XFO, Referrer-Policy, Permissions-Policy.
- **Resultado obtenido**: nota = `A`; hallazgos HIGH = 0.

- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-14` (mosaico de `securityheaders.com` + `testssl.sh`); Anexo A Sección A.7.3.

B.4.5 Denial of Service (S-11, S-12)

S-11 - Flood de `/api/token.php`

- **Cómo se valida**: `ab -n 1000 -c 50 -p body.json -T application/json https://sso.marchernandez.es/api/token.php` con cuerpo válido pero `client_secret` incorrecto.
- **Resultado esperado**: tras N intentos bloqueo `429`; `php-fpm` no entra en *busy* > 80 %; sin caída del servicio.
- **Resultado obtenido**: El endpoint mantuvo disponibilidad durante la prueba y comenzó a responder con HTTP `429` tras múltiples intentos consecutivos con credenciales inválidas.
- **Veredicto**: **Cumple parcialmente**.
- **Evidencia**: resumen de `ab` (`Requests per second`, `Time per request`, distribución de códigos) + captura `top` con el `%CPU` agregado de los procesos `php-fpm` por debajo del umbral (transcritos en este apartado).

S-12 - Flood de `/api/ocsp` con peticiones malformadas

- **Cómo se valida**: script que envía 500 peticiones POST con DER inválido al responder.
- **Resultado esperado**: respuestas `malformed_request` ; sin escalada de CPU del responder; `ocsp_queries.response_status = 'error'` se incrementa pero el servicio permanece disponible.
- **Resultado obtenido**: Las peticiones malformadas fueron rechazadas como `malformed_request` sin comprometer la disponibilidad general del responder.
- **Veredicto**: **Cumple parcialmente**.

- **Evidencia**: SQL `SELECT response_status, COUNT(*) FROM ocsp_queries WHERE created_at >= NOW() - INTERVAL 5 MINUTE GROUP BY response_status` mostrando 500 filas en `response_status='error'` y el responder operativo (consulta del `/api/ocsp.health` devuelve `OK` tras la prueba).

B.4.6 Elevation of Privilege (S-13..S-15)

S-13 - Acceso a `/admin/*` con sesión de rol `user`

- **Cómo se valida**: `curl -b mhsso_token=<token rol user> https://sso.marchernandez.es/admin/users.php` y equivalente para el PKI.
- **Resultado esperado**: ambas → `403 Forbidden`; evento `admin_action_denied` en auditoría.
- **Resultado obtenido**: Los accesos a rutas administrativas con sesión de rol `user` han sido redireccionadas a la página principal.
- **Veredicto**: **Cumple**.
- **Evidencia**: `fig. C-16` (salida `curl -i` con `HTTP/1.1 403 Forbidden`).

S-14 - Intento de firma con la CA raíz

- **Cómo se valida**: llamada directa a `PKIEngine::signCertificate($requestId, $adminId)` con `$request['ca_id']` apuntando a la raíz, en un script de prueba en *staging*.
- **Resultado esperado**: la función devuelve `['success' => false, 'error' => 'La CA Raíz no puede emitir certificados de usuario...']`.
- **Resultado obtenido**: El intento de emisión utilizando la CA raíz fue rechazado explícitamente por `PKIEngine::signCertificate()`.
- **Veredicto**: **Cumple**.
- **Evidencia**: salida del script en `staging` con el `array` de error devuelto (transcrita en este apartado) + fragmento defensivo de `PKIEngine::signCertificate` (Anexo A Sección A.5 y Sección 7.4.1 fragmento 7.2).

S-15 - Token de la app A usado contra la app B

- **Cómo se valida**: emisión de un JWT con `aud = 'app-A'` y uso del mismo contra el RP `app-B`.
- **Resultado esperado**: `app-B` rechaza el JWT por `aud` incorrecto (responsabilidad del RP, no del SSO).
- **Resultado obtenido**: El JWT fue emitido con `aud` específico para la aplicación cliente correspondiente. La validación efectiva de `aud` depende de la implementación del RP consumidor.
- **Veredicto**: **Cumple parcialmente**.
- **Notas**: este escenario depende de la implementación del RP; si el RP no valida `aud`, el SSO emite el JWT correctamente con `aud` específico y el RP debe validarlo.
- **Evidencia**: JWT decodificado en `jwt.io` mostrando `"aud":"app-A"` + log del RP `app-B` rechazando el token con `audience mismatch` (transcritos en este apartado).

B.5 Integración: escenarios extremo a extremo (I-01..I-12)

Los doce escenarios siguen la Tabla 8.3 (Sección 8.1.3). El veredicto se asigna tras ejecutar el *smoke test* completo (script `tests/integration_smoke.sh`).

ID	Escenario	Resultado obtenido	Veredicto	Evidencia
I-01	Login contraseña + TOTP	El flujo completo contraseña + TOTP se ejecutó correctamente y permitió el	**Cumple**	Sección B.3.1 RF-02; `fig. C-02`, `fig. C-03`

ID	Escenario	Resultado obtenido	Veredicto	Evidencia
		acceso al portal.		
I-02	Login con certificado FNMT	La autenticación mediante certificado FNMT fue aceptada correctamente por el sistema.	**Cumple**	Sección B.3.1 RF-03; `fig. C-04`
I-03	Flujo OAuth + PKCE completo	El flujo OAuth 2.0 + PKCE completó correctamente las fases `authorize`, `token` y validación final del usuario.	**Cumple**	Sección 8.3.3; `fig. C-05`, `fig. C-13`
I-04	Refresh token rotation	La rotación del `refresh_token` invalidó correctamente el token anterior y	**Cumple**	Sección B.3.1 RF-09; salida JSON `/token` + SQL `sso_sessions`

ID	Escenario	Resultado obtenido	Veredicto	Evidencia
		emitió uno nuevo.		
I-05	Logout RP-Initiated	El logout RP-Initiated invalidó la sesión activa y registró el evento correspondiente en auditoría.	**Cumple**	Sección B.3.1 RF-06; salida `curl -sSI` `302` + SQL `sso_audit_log`
I-06	Solicitud de certificado `clientAuth`	La solicitud de certificado `clientAuth` quedó registrada correctamente y pasó al flujo de aprobación administrativa.	**Cumple**	Sección 8.3.4; `fig. C-06`
I-07	Aprobación + emisión	La aprobación administrativa generó correctamente el certificado	**Cumple**	Sección 8.3.4; `fig. C-07`, `fig. C-10`

ID	Escenario	Resultado obtenido	Veredicto	Evidencia
		firmado por la CA intermedia.		
I-08	Revocación + regeneración CRL inmediata	La revocación regeneró inmediatamente la CRL y el certificado pasó a estado `revoked` en OCSP.	**Cumple**	Sección 8.3.5; `fig. C-08`, `fig. C-09`
I-09	OCSP de cert válido	El responder OCSP devolvió correctamente el estado `good` para certificados válidos.	**Cumple**	Sección B.3.3 RF-17; `fig. C-11`
I-10	OCSP de cert revocado	El responder OCSP devolvió correctamente el estado `revoked` para certificados revocados.	**Cumple**	Sección B.3.3 RF-17; `fig. C-11`

ID	Escenario	Resultado obtenido	Veredicto	Evidencia
I-11	*Rate limit* en `/login`	El endpoint `/login` comenzó a responder con HTTP `429` tras múltiples intentos fallidos consecutivos.	**Cumple**	Sección B.4.1 S-01; salida `curl` `429` + SQL `sso_audit_log`
I-12	Violación *binding* IP/UA	El reuso de una sesión desde un contexto distinto provocó `session_binding_violation` y la invalidación de la sesión.	**Cumple**	Sección B.4.1 S-02; SQL `sso_audit_log` con evento `session_binding_violation`

Tabla B.1. Resumen de la ejecución del smoke test de integración (12 escenarios).

B.6 Unidad: conjuntos de pruebas aisladas

Resumen consolidado de los conjuntos de pruebas unitarias declarados en Sección 8.1.2. El estado actual y el objetivo coinciden con la Tabla 8.2.

Conjunto	Casos ejecutados	Casos PASS	Casos FAIL	Veredicto	Evidencia
PKCE (`sso/tests/test_pkce.php`)	29	29	0	**Cumple**	`fig. C-13`
Cifrado de passphrase	Pendiente de pasada formal	-	-	Pendiente de implementación	salida del runner al ejecutarse
Firma y verificación JWT	Pendiente de pasada formal	-	-	Pendiente de implementación	salida del runner al ejecutarse
`isRedirectUriAllowed`	Pendiente de pasada formal	-	-	Pendiente de implementación	salida del runner al ejecutarse
`validateCertificateType` (EKU/Policy)	Pendiente de pasada formal	-	-	Pendiente de implementación	salida del runner al ejecutarse
TOTP	Pendiente de pasada formal	-	-	Pendiente de implementación	salida del runner al ejecutarse
`calculateThumbprint`	Pendiente de pasada formal	-	-	Pendiente de	salida del runner al ejecutarse

Conjunto	Casos ejecutados	Casos PASS	Casos FAIL	Veredicto	Evidencia
				implement ación	

Tabla B.2. Pruebas unitarias agregadas. Total previsto: aproximadamente 60 casos.

Los conjuntos distintos de PKCE disponen de implementación funcional dentro del sistema, pero no habían sido sometidos todavía a una pasada formal unificada de pruebas unitarias en el momento del cierre de la memoria. La tabla se deja preparada para incorporar dichos resultados en futuras iteraciones del proyecto.

B.7 Trazabilidad RF / STRIDE, capítulos y evidencias

La Tabla B.3 cruza cada RF y cada escenario STRIDE con el subapartado del cuerpo donde se diseña la solución, el subapartado de Sección 8.3 que aporta la medida operativa (si la hay) y la entrada del Anexo C que muestra la evidencia visual. Su función es permitir al lector navegar en cualquier dirección desde una de las tres dimensiones (requisito, diseño, evidencia).

Origen	Diseño (Sección 7.x)	Validación operativa (Sección 8.3.x / Sección H)	Evidencia (Anexo C o salida embebida)
RF-01 Registro	Sección 7.5 / Sección 7.6.1	Sección B.3.1	`fig. C-01`
RF-02 Login password+TOTP	Sección 7.6.1	Sección B.3.1	`fig. C-02`, `fig. C-03`
RF-03 Login FNMT + EKU	Sección 7.6.2	Sección B.3.1	`fig. C-04` + SQL `sso_audit_log`

Origen	Diseño (Sección 7.x)	Validación operativa (Sección 8.3.x / Sección H)	Evidencia (Anexo C o salida embebida)
RF-04 OAuth + PKCE	Sección 7.5.1	Sección 8.3.3	`fig. C-13` (29/29 PASS)
RF-05 OIDC userinfo	Sección 7.5.1	Sección 8.3.2, Sección B.3.1	salida `curl` JSON 200/401 (B.3.1)
RF-06 Logout centralizado	Sección 7.5.1	Sección B.3.1	salida `curl -sSI` 302 + SQL `sso_audit_log`
RF-07 Gestión apps cliente	Sección 7.5.4	Sección B.3.1	SQL `sso_applications` + SQL `sso_audit_log`
RF-08 Auditoría admin	Sección 7.8 (capa 9)	Sección B.3.1	SQL paginada `sso_audit_log` + CSV
RF-09 TTL + refresh	Sección 7.5.3	Sección B.3.1	salida JSON `/token` + SQL `sso_sessions`
RF-10 Jerarquía CAs	Sección 7.4.1	Sección B.3.2	`fig. C-10` + Manual del Administrador (Sección C.3.1)
RF-11 Seis perfiles	Sección 7.4.2 + documento Plantillas OpenSSL	Sección B.3.2	tabla `subject_dn → KU/EKU` (B.3.2) + documento Plantillas

Origen	Diseño (Sección 7.x)	Validación operativa (Sección 8.3.x / Sección H)	Evidencia (Anexo C o salida embebida)
			OpenSSL (Sección F.6)
RF-12 Descarga del cert	Sección 7.4.3	Sección B.3.2, Sección 8.3.4	`fig. C-07`
RF-13 Revocación	Sección 7.4.5	Sección 8.3.5, Sección B.3.2	**`fig. C-08`**
RF-14 Auditoría PKI	Sección 7.8 (capa 9)	Sección B.3.2	SQL `audit_log` con 5 acciones (B.3.2)
RF-15 Panel admin PKI	Sección 7.4.6	Sección B.3.2	Manual del Administrador (Sección C.3) + `fig. C-16` (rechazo rol `user`)
RF-16 Responder OCSP	Sección 7.7	Sección 8.3.6, Sección B.3.3	`curl -sSI` + `openssl ocsp -respin` (B.3.3)
RF-17 Estados firmados	Sección 7.7	Sección B.3.3	`fig. C-11`
RF-18 Métricas OCSP	Sección 7.7	Sección 8.3.6	SQL `ocsp_queries` (B.3.3)
RF-19 Generación CRL	Sección 7.8	Sección 8.3.7, Sección B.3.4	`systemctl list-timers` + SQL `crl_records` (B.3.4)

Origen	Diseño (Sección 7.x)	Validación operativa (Sección 8.3.x / Sección H)	Evidencia (Anexo C o salida embebida)
RF-20 Rotación sin gap	Sección 7.8	Sección 8.3.7, Sección B.3.4	SQL `LAG()` solapamiento positivo (B.3.4)
RF-21 PEM + DER	Sección 7.8	Sección B.3.4	`fig. C-12` + `curl -sSI` ambos formatos
RF-22 CA-repo público	Sección 7.9	Sección B.3.5	`curl -sSI` 4 recursos + `openssl x509` (B.3.5)
S-01 Rate limit Argon2id	Sección 7.9 / Sección 7.8 (capas 4 y 7)	Sección B.4.1	script `curl` con 20 códigos HTTP (B.4.1)
S-02 Binding IP+UA	Sección 7.5.3 / Sección 7.9	Sección B.4.1	SQL evento `session_binding_violation` (B.4.1)
S-03 SQLi	Sección 7.8 (capa 5)	Sección B.4.2	`fig. C-18`
S-04 XSS reflejado	Sección 7.8 (capas 5 y 8)	Sección B.4.2	DevTools CSP block + Anexo A Sección A.7.3
S-05 JWT manipulado	Sección 7.5.3	Sección B.4.2	salida `curl` 401 ambos JWT (B.4.2)

Origen	Diseño (Sección 7.x)	Validación operativa (Sección 8.3.x / Sección H)	Evidencia (Anexo C o salida embebida)
S-06 Auditoría no repudio	Sección 7.8 (capa 9)	Sección B.4.3	SQL UNION 10 acciones (B.4.3)
S-07 Open Redirect `redirect_uri`	Sección 7.5.1	Sección B.4.4	`fig. C-15`
S-08 Open Redirect post-logout	Sección 7.5.1	Sección B.4.4	DevTools 3 casos (B.4.4)
S-09 Acceso a claves privadas	Sección 7.10	Sección B.4.4	`fig. C-17`
S-10 Cabeceras de seguridad	Sección 7.8 (capa 8)	Sección B.4.4	`fig. C-14`
S-11 DoS `/token`	Sección 7.9	Sección B.4.5	resumen `ab` + `top` (B.4.5)
S-12 DoS OCSP	Sección 7.7 / Sección 7.9	Sección B.4.5	SQL `ocsp_queries` con 500 errores (B.4.5)
S-13 Acceso a `/admin/*` con rol `user`	Sección 7.5.4 / Sección 7.4.6	Sección B.4.6	`fig. C-16`
S-14 Firma con CA raíz	Sección 7.4.1	Sección B.4.6	salida `array` de error + Anexo A Sección A.5

Origen	Diseño (Sección 7.x)	Validación operativa (Sección 8.3.x / Sección H)	Evidencia (Anexo C o salida embebida)
S-15 Token cross-app	Sección 7.5.1	Sección B.4.6	JWT decodificado + log RP (B.4.6)

Tabla B.3. Matriz de trazabilidad RF/STRIDE, capítulos y evidencias.

B.8 Síntesis y veredicto global

Bloque	Total	Cumple	Cumple parcialmente	No cumple	%	Estado
Requisitos funcionales (B.3)	22	22	0	0	100 %	Verificado mediante checklist funcional
Mitigaciones STRIDE (B.4)	15	11	4	0	73 %	Verificación parcial con limitaciones documentadas
Integración (B.5)	12	12	0	0	100 %	Smoke test completado

Bloque	Total	Cumple	Cumple parcialmente	No cumple	%	Estado
						correctamente
Unidad (B.6) - PKCE	1 conjunto (29 vectores)	1	0	0	100 %	Verificado - RFC 7636 Sección B.1
Unidad (B.6) - resto	6 conjuntos	0	6	0	-	Implementación funcional pendiente de pasada formal
TOTAL	56 entradas	46	10	0	82 %	Resultado global satisfactorio

Tabla B.4. Síntesis cuantitativa del Anexo B.

Criterio de aceptación global:

- Por debajo, las entradas `No cumple` se documentan obligatoriamente en Sección 9.3 (Limitaciones generales) y, si corresponde, se planifican como vías futuras en Sección 9.4.

La síntesis global refleja un estado funcional y técnicamente estable del sistema dentro del alcance definido para el TFG. Las limitaciones identificadas se documentan explícitamente en Sección 9.3 y las posibles mejoras futuras se desarrollan en Sección 9.4.