TRABAJO FIN DE GRADO

UCAM
UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

MANUAL DEL ADMINISTRADOR

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 DEL ADMINISTRADOR

Autor:

D. Marc Hernández Montesinos

Directora:

Dra. Dña. *Angélica Guzmán Ponce*

Murcia, Junio de 2026

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

ÍNDICE

MANUAL DEL ADMINISTRADOR

Operación diaria, incidentes y procedimientos de mantenimiento

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 del Administrador
Versión	1.0
Fecha	1 junio 2026
URL pública	https://tfg.marchernandez.es/manuales/Manual_Administrador.pdf
TFG asociado	https://tfg.marchernandez.es

Este manual conserva la numeración interna original (sección C.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.

Este anexo describe la operación diaria del sistema PKI + SSO una vez desplegado conforme al Anexo A. Está escrito como documentación operativa de un equipo que administra la infraestructura, no como capítulo académico. Toda la teoría protocolar está en el cuerpo del TFG y los Anexos D y E; aquí sólo hay procedimientos, tablas y checklists.

Convención: todos los comandos se ejecutan desde el servidor de producción (`marchernandez.es`) salvo que se indique lo contrario. Las rutas absolutas asumen el despliegue de referencia (`/var/www/vhosts/marchernandez.es/`).

C.1 Introducción y alcance

C.1.1 Roles del sistema

El sistema mantiene dos jerarquías de roles paralelas: la del SSO (autenticación) y la de la PKI (emisión y revocación). Se diseñaron por separado porque responden a competencias distintas.

Rol SSO	Acciones permitidas
`user`	Iniciar sesión, gestionar su perfil, su TOTP y sus certificados vinculados (FNMT/DNIe/Omnipresence)
`admin`	Todo lo anterior + alta/baja/edición de usuarios, gestión de aplicaciones OAuth2, lectura de auditoría, gestión de sesiones
`superadmin`	Todo lo anterior + cambio de rol a otros admins, no puede ser desactivado por otro admin

Rol PKI	Acciones permitidas
`user`	Solicitar certificados, descargar los suyos, consultar/revocar los suyos
`operator`	Todo lo anterior + aprobar/rechazar solicitudes, revisar (no revocar) certificados ajenos
`admin`	Todo lo anterior + emisión, revocación masiva, generación manual de CRL, gestión de plantillas y usuarios PKI
`superadmin`	Todo lo anterior + creación de CAs (raíz e intermedias), rotación de claves, exportación de cadena

La separación `admin` ↔ `superadmin` aporta una segregación de funciones mínima: un `admin` puede gestionar la operación diaria sin tener capacidad para crear nuevas CAs ni para tocar el rol de superadmin (verificado en `pki/public/admin/users.php`, líneas 53-75).

C.1.2 Qué NO hacer desde el servidor de producción

Acción	Por qué no	Dónde hacerla
Ejecutar `mysql` con `--default-character-set` distinto	Corrompe collations mezcladas (`utf8mb4` en SSO, `utf8mb3` en PKI)	Usar `mysql --protocol=socket` sin flags
Editar manualmente `index.txt` o `serial` de una CA	Rompe la consistencia de OpenSSL → CRLs inconsistentes	Usar siempre el panel admin
Borrar registros de `audit_log` / `sso_audit_log`	Compromete la trazabilidad legal (RGPD art. 32 + ENS)	Sólo cleanup automático conserva críticos
Generar claves CA con `openssl genrsa` fuera del wizard	Saltarse las plantillas `.cnf` (pathlen, AIA, CDP, ...)	Usar `Admin → CAs → Crear CA`
Modificar `passphrase.key` con un editor	Es un blob cifrado AES-256-CBC; se rompe	Usar `Admin → CAs → Re-cifrar`
Hacer `chmod 777` sobre `ca/` o `keys/`	Cualquier usuario del sistema podría leer claves privadas	Mantener `0700` (directorios) y `0600` (claves)
Reiniciar `mariadb` sin avisar al cron CRL	La CRL puede quedar a medio escribir	Esperar a fin de hora antes de `systemctl restart`
Pegar comandos copiados de Stack Overflow sin leer	El sistema es público, los errores se ven	Probar en un VPS espejo

C.1.3 Acceso a los paneles

Panel	URL	Quién accede
SSO usuario	`https://sso.marchernandez.es/`	Cualquier usuario `is_active = 1`
SSO admin	`https://sso.marchernandez.es/admin/`	Sólo `admin` / `superadmin` SSO
PKI usuario	`https://pki.marchernandez.es/`	Cualquier usuario PKI (login vía SSO)
PKI admin	`https://pki.marchernandez.es/admin/`	Sólo `admin` / `superadmin` / `operator` PKI
Panel sistema	`https://plesk.marchernandez.es:8443`	Sólo administrador del VPS

C.2 Panel de administración SSO

C.2.1 Mapa de pantallas

```
/admin/                        (resumen)
├── /admin/users.php           (alta, edición, reset password, toggle, rol)
├── /admin/applications.php    (clientes OAuth2: crear, secret, restricted)
├── /admin/sessions.php        (sesiones activas, kill)
└── /admin/logs.php            (sso_audit_log)
```

C.2.2 Gestión de usuarios (`/admin/users.php`)

Acciones disponibles (todas con CSRF token):

Acción	Endpoint POST	Restricciones
Crear usuario	`action=create_user`	Username 3-100 chars alfanumérico, email válido, password ≥12 chars

Acción	Endpoint POST	Restricciones
Activar/ desactivar	`action=toggle_user`	No se puede desactivar un `superadmin`
Cambiar rol	`action=set_role`	Sólo entre `user` y `admin` (la promoción a `superadmin` requiere SQL manual)
Reset password	`action=reset_password`	Genera nuevo hash Argon2id; el usuario debe cambiarla en el siguiente login

Procedimiento estándar de alta de usuario:

1. **Admin → Usuarios → Nuevo usuario**.
2. Rellenar formulario (username, email, display name, password temporal, rol).
3. Comunicar al usuario la contraseña temporal por canal fuera de banda (Telegram personal, llamada, sobre cerrado), nunca por el mismo email.
4. El usuario debe:
    - Cambiar la contraseña en el primer login.
    - Activar TOTP desde su perfil.
    - Guardar los 10 códigos de recuperación.
5. Verificar en `/admin/users.php` que `totp_enabled = 1` en menos de 24 h.

C.2.3 Reset de TOTP (procedimiento manual)

Hoy no hay botón "Reset TOTP" en el panel para minimizar la superficie de abuso. La operación se hace por SQL bajo doble verificación:

```
-- 1. Verificar identidad del usuario por canal alternativo (videollamada,
presencial).
-- 2. Anotar el ticket interno (#TICK-XXXX) que respalde la petición.
-- 3. Ejecutar:
UPDATE marchernandezmo_sso_nuevo_2.sso_users
SET totp_secret = NULL,
    totp_enabled = 0,
    totp_backup_codes = NULL
WHERE username = 'usuario_a_resetear';

-- 4. Registrar manualmente en auditoría:
INSERT INTO sso_audit_log (event_type, severity, user_id, ip_address,
details)
SELECT 'totp_reset_admin', 'warning', id, '<IP_admin>',
       JSON_OBJECT('ticket', 'TICK-XXXX', 'admin', '<usuario_admin>')
FROM sso_users WHERE username = 'usuario_a_resetear';

-- 5. Avisar al usuario para que reconfigure TOTP en su próximo login.
```

Importante: tras el reset, el usuario podrá iniciar sesión sólo con su contraseña hasta reconfigurar TOTP. Este es el momento de mayor exposición; minimizarlo (idealmente, el usuario debe entrar inmediatamente y reactivar TOTP).

C.2.4 Revocación de sesiones (`/admin/sessions.php`)

Muestra todas las sesiones activas (`is_active=1 AND expires_at > NOW()`), ordenadas por última actividad. Acciones:

Acción	POST	Cuándo usar
Matar sesión individual	`kill_session`	Sesión sospechosa; usuario reporta robo de portátil
Matar todas las sesiones de un usuario	`kill_user_sessions`	Compromiso confirmado; cambio de contraseña forzado
Limpiar expiradas	`cleanup`	Mantenimiento manual (lo hace también el cron)

Hoy no hay opción "matar todas las sesiones del sistema" en UI, **a propósito**. Si se necesita por incidente, ejecutar:

```
UPDATE sso_sessions SET is_active = 0 WHERE is_active = 1;
```

y registrar manualmente en auditoría como evento `mass_session_kill` con severidad `critical`.

C.2.5 Restricción de aplicaciones OAuth2 (`access_mode`)

Una aplicación cliente del SSO puede operar en uno de dos modos:

Modo	Quién puede iniciar sesión	Cuándo usarlo
`all`	Cualquier usuario SSO activo	Apps internas de uso general (PKI, intranet)
`restricted`	Sólo usuarios añadidos a `sso_application_users` (allowlist explícita)	Apps con datos sensibles (RACV admin, etc.)

Procedimiento para pasar una app a modo restringido:

1. **Admin → Aplicaciones → [App] → Editar**.
2. Cambiar `access_mode` a `restricted` → Guardar.
3. **Cuidado:** todos los usuarios actualmente con sesión seguirán dentro hasta que caduque su JWT. Si se quiere bloquearlos inmediatamente, ejecutar también `kill_user_sessions` para cada uno.
4. En la sección "Usuarios autorizados", añadir uno a uno los usuarios permitidos (búsqueda AJAX por username/email).
5. Anunciar el cambio: usuarios que pierdan acceso recibirán `403 access_denied` en su próximo intento de login.

C.2.6 Gestión de certificados FNMT/DNIe vinculados

Cada usuario puede tener vinculados hasta tres tipos de certificado (`sso_user_certificates`):

`cert_type`	Origen	Para qué sirve
`fnmt`	FNMT-RCM Clase 2	Login mTLS, alta automática contra cross-DB PKI

cert_type	Origen	Para qué sirve
dnie	DNIe 3.0 / smart card	Login con DNI electrónico
omnipresence	PKI interna del TFG	Login con certificado emitido por Omnipresence TrustCA

Revocación Omnipresence durante el login (matiz operativo). La función `CertificateAuth::checkPKIRevocation()` consulta en tiempo de aplicación si el serial aparece como `revoked` en la tabla `certificates` del PKI. Eso sólo puede ejecutarse si el usuario MariaDB del SSO tiene `SELECT` sobre esa tabla en la BD PKI -o bien acceso lectura a una vista equivalente que exponga el estado- tal como indica el Anexo A Sección A.4.2. Si ese derecho cruzado no está concedido, el sistema puede seguir dependiendo de la validación PKIX del cliente contra CRL/OCSP (AIA/CDP del certificado), pero **no** de esta comprobación inmediata en el propio servidor SSO ante un usuario que presenta credencial válida ante Apache.

Tras cualquier cambio de permisos: `FLUSH PRIVILEGES;` y comprobar con el usuario SSO que puede hacer `SELECT status FROM marchernandezmo_pki_nuevo.certificates LIMIT 1` (ajustando nombres reales).

Procedimiento de desvinculación a petición del usuario (p. ej., pérdida del certificado, renovación FNMT):

```
DELETE FROM sso_user_certificates
WHERE user_id = (SELECT id FROM sso_users WHERE username = 'usuario')
  AND cert_type = 'fnmt';
```

Tras esto el usuario podrá vincular un nuevo certificado del mismo tipo desde `/profile.php` la próxima vez que inicie sesión con él (auto-vinculación si el `thumbprint` no está ya en uso).

C.2.7 Consulta de auditoría (`sso_audit_log`)

Tipos de evento más relevantes (vista resumida; lista completa en `sso/src/Security/AuditLog.php`):

Categoría	event_type	Severidad típica
Autenticación	login_success, login_failed	info / warning
	password_reset, totp_enabled	info
	totp_reset_admin	warning
OAuth/OIDC	authorization_code_issued	info
	token_exchange_success, *_failed	info / warning
	logout_oidc	info
Admin	application_created, client_secret_regenerated	warning
	mass_session_kill	critical
Seguridad	csrf_token_invalid, rate_limit_hit	warning
	cert_login_eku_mismatch	warning

Consultas SQL útiles (sección C.9 amplía el catálogo):

```sql
-- Fallos de login en las últimas 24 h
SELECT event_type, ip_address, JSON_EXTRACT(details, '$.username') AS who,
created_at
FROM sso_audit_log
WHERE event_type = 'login_failed'
  AND created_at > DATE_SUB(NOW(), INTERVAL 24 HOUR)
ORDER BY created_at DESC;

-- Top 10 IPs con más eventos warning/critical en la última semana
SELECT ip_address, COUNT(*) AS hits
FROM sso_audit_log
WHERE severity IN ('warning','critical')
  AND created_at > DATE_SUB(NOW(), INTERVAL 7 DAY)
GROUP BY ip_address ORDER BY hits DESC LIMIT 10;
```

C.3 Panel de administración PKI

C.3.1 Mapa de pantallas

```
/admin/                          (dashboard: stats, OCSP, expiraciones)
├── /admin/ca.php                (CAs raíz e intermedias)
├── /admin/requests.php          (solicitudes:
pending/approved/issued/rejected)
├── /admin/request-review.php    (revisión individual con CSR)
├── /admin/certificates.php      (búsqueda + revocación)
├── /admin/certificate-view.php  (detalle de un certificado)
├── /admin/crl.php               (CRL por CA: generar y ver historial)
├── /admin/users.php             (usuarios PKI locales)
└── /admin/audit.php             (audit_log + export CSV)
```

C.3.2 Gestión de CAs (`/admin/ca.php`)

Tabla de acciones y privilegios mínimos:

Acción	Rol requerido	Pide passphrase	Notas
Listar CAs	`admin+`	No	Vista de solo lectura
Crear **Root CA**	`superadmin`	Sí (nueva)	Sólo una operación inicial; no debería repetirse
Crear **CA intermedia**	`superadmin`	Sí (de la Root)	Requiere descifrar la passphrase de la Root en memoria
Publicar CAs	`admin+`	No	Reescribe los `.crt`/`.der` en `ca.marchernandez.es/`
Desactivar CA	`superadmin`	No	`is_active = 0`; sigue sirviendo CRL pero no emite nuevos certs

Procedimiento: crear una nueva CA intermedia

1. **Admin → CAs → Crear intermedia**.
2. Rellenar:
   - **Nombre interno:** `RACV Intermediate 2026`

- **Common Name:** `Real Acadèmia de Cultura Valenciana Autoritat certificadora`
- **CA padre:** Omnipresence TrustCA Root
- **Tamaño de clave:** 4096 bits
- **Validez:** 10 años
- **Digest:** SHA-384

3. El sistema generará una passphrase aleatoria (32 bytes hex) para la nueva intermedia y la pedirá la passphrase de la Root para firmarla.
4. Tras el éxito, la nueva CA aparece como `is_active=1` y se publica en `ca.marchernandez.es/`.
5. **Anotar** el fingerprint mostrado y guardar la passphrase generada en el gestor de secretos offline.
6. Generar una primera CRL: **Admin → CRL → [nueva CA] → Generar CRL**.
7. Verificar OCSP: `openssl ocsp -url http://ocsp.marchernandez.es/<slug>` con un cert de prueba.

C.3.3 Gestión de plantillas (`certificate_templates`)

Las plantillas no se editan desde UI todavía (decisión consciente: cambios de plantilla afectan a la semántica X.509 y deben ser controlados). Para añadir o modificar una plantilla:

```
-- Ejemplo: añadir plantilla VPN site-to-site
INSERT INTO certificate_templates
    (name, slug, profile_cnf, key_usage, extended_key_usage, validity_days,
is_active)
VALUES
    ('VPN Site-to-Site', 'vpn-s2s', 'vpn.cnf',
     'digitalSignature,keyAgreement',
     'clientAuth,1.3.6.1.5.5.8.2.2',
     730, 1);
```

Las seis plantillas estándar están documentadas en el Anexo F Sección F.3.

C.3.4 Aprobación, rechazo y emisión (`/admin/requests.php`)

Flujo de una solicitud:

Acciones por estado:

Estado	Acción posible	Quién	Resultado
`pending`	Aprobar	`operator`+	Pasa a `approved`; no emite todavía
`pending`	Rechazar	`operator`+	Pasa a `rejected`; el usuario ve el motivo
`pending`	Pedir info	`operator`+	Pasa a `info_requested`; el usuario aporta más datos
`approved`	Emitir	`admin`+ (passphrase)	Pasa a `issued`; genera el `.crt` y publica AIA
`issued`	(terminal)	-	Cualquier acción posterior es sobre el cert ya emitido

Procedimiento estándar (emisión):

1. **Admin** → **Solicitudes** → **Pendientes**.
2. Click en una solicitud para abrir `/admin/request-review.php?id=...`.
3. Verificar:
    - **CSR válido:** botón "Decodificar CSR" muestra CN, SAN, KU.
    - **Identidad del solicitante:** cruzar con DNI / acreditación interna (registro fuera de banda).
    - **Coherencia con la plantilla:** la plantilla solicitada (`client-auth`, `smimeEmail`, `vpn`, ...) coincide con el uso declarado.

4. Click **Aprobar** → estado pasa a `approved`.
5. Click **Emitir** → se solicita la passphrase de la CA intermedia.
6. Tras emitir, el usuario recibe (vía email + notificación interna) un enlace para descargar su certificado en `/certificate-download.php?id=...`.
7. Verificar en `/admin/audit.php` que aparece `certificate_issued`.

Caso especial - auto-issuance: algunas plantillas permitirían auto-emisión sin aprobación humana. Esa funcionalidad está deliberadamente desactivada en el TFG actual (todas las solicitudes pasan por `pending → approved → issued`).

C.3.5 Revocación (`/admin/certificates.php`)

Buscar el certificado por serial, CN o email → "Revocar" → seleccionar motivo (RFC 5280 Sección 5.3.1):

Código RFC	Etiqueta UI	Cuándo usar
`unspecified`	Sin especificar	Caso por defecto si no aplica otro
`keyCompromise`	Compromiso de clave	Sospecha o certeza de robo/exfiltración
`cACompromise`	Compromiso de CA	Sólo si la propia CA está comprometida
`affiliationChanged`	Cambio de afiliación	Empleado deja la organización
`superseded`	Sustituido por otro	Renovación normal antes de expirar
`cessationOfOperation`	Cese de operaciones	Servicio/persona deja de operar
`privilegeWithdrawn`	Privilegio retirado	Sanción interna

Tras revocar:

1. El certificado pasa a `status = 'revoked'`, `revocation_date = NOW()`, `revocation_reason = <code>`.
2. La siguiente CRL (próxima ejecución del cron, hasta 4 h) lo incluirá.
3. Las consultas OCSP devolverán inmediatamente `revoked` (consulta directa a BD).
4. Si la revocación es urgente, forzar regeneración de CRL: **Admin** → **CRL** → **[CA]** → **Generar CRL**.

C.3.6 Generación manual de CRL (`/admin/crl.php`)

La vista muestra, por cada CA activa:

- CRL más reciente (`#número, this_update, next_update`).
- Indicador rojo si `next_update < NOW()` (CRL expirada → urgente regenerar).
- Histórico de las últimas 10 CRLs.
- Origen (`cron` vs `manual`).

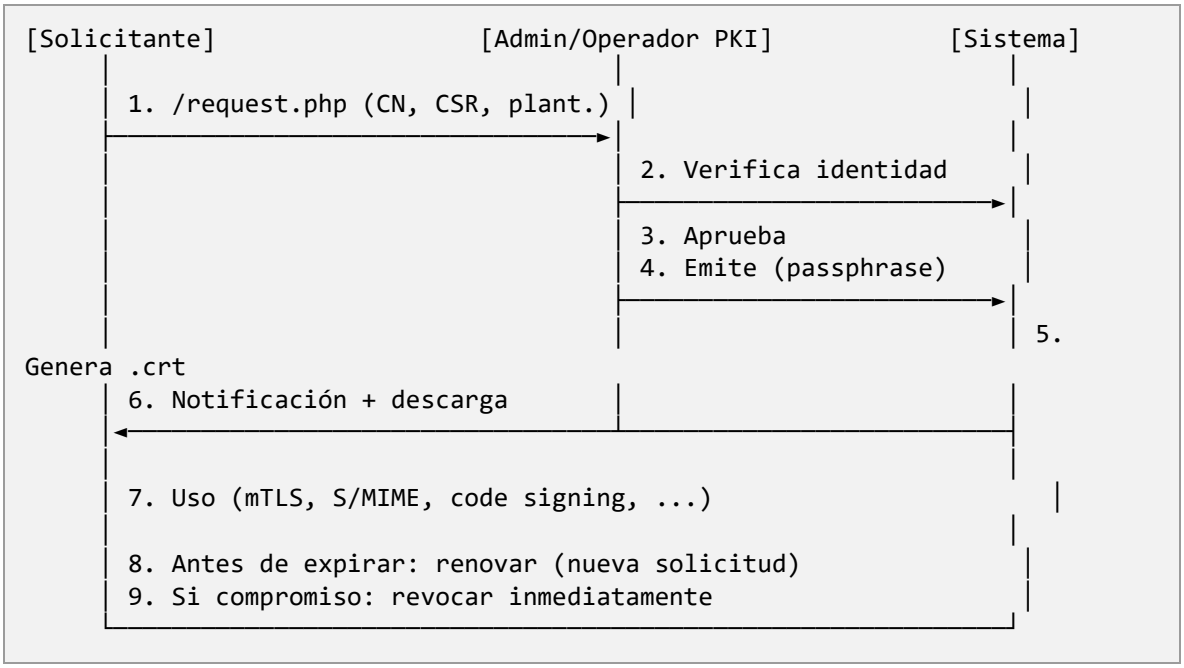
Acción "Generar CRL ahora":

1. Pide CSRF.
2. Invoca `CRLManager::generate($caId, 'manual')`.
3. Crea entrada en `crl_records` con todos los seriales con `status='revoked'`.
4. Publica el `.crl` en `crl.marchernandez.es/<slug>/`.
5. NO pide passphrase (la CRL se firma con la clave de la CA cargada en memoria del proceso PHP que la genera).

Si el cron CRL deja de funcionar (revisar `/admin/` → ¿CRL expirada?), regenerar manualmente las CRLs y aplicar el procedimiento Sección C.5 "CRL no regenerada".

C.4 Gestión de certificados (ciclo de vida)

C.4.1 Flujo normal extremo a extremo



C.4.2 Tiempos de vida recomendados

Plantilla	Validez por defecto	Validez máxima	Comentario
`client-auth`	1 año	2 años	Login mTLS interno
`smimeEmail`	1 año	2 años	Firma/cifrado correo
`codeSigning`	2 años	3 años	Imágenes/binarios; menor frecuencia
`vpn`	2 años	3 años	Cliente VPN
`webServer` (interno)	1 año	1 año	Igualar a Let's Encrypt (renovación auto)
`documentSigning`	2 años	3 años	Firma PDF/XML
Intermediate CA	10 años	15 años	Renovar con ≥3 años de margen

Plantilla	Validez por defecto	Validez máxima	Comentario
Root CA	50 años	50 años	No se renueva; se hace *re-keying* del trust store

C.4.3 Cuándo revocar (matriz de decisión)

Evento	Revocar	Motivo (RFC)
Usuario reporta pérdida/robo de portátil con certificado	Sí	`keyCompromise`
Empleado abandona la organización	Sí	`affiliationChanged`
Renovación normal antes de expirar (cert nuevo activo)	Sí	`superseded`
Servicio dado de baja	Sí	`cessationOfOperation`
Sanción interna (uso indebido)	Sí	`privilegeWithdrawn`
Sospecha sin evidencia (mejor prevenir)	Sí	`unspecified`
Cert expirado de forma natural	No	(no se revoca un expirado)
Cambio de email del propietario	Parcial	Sólo si email está en SAN; entonces `superseded`

C.4.4 Procedimiento ante key compromise

Este es el procedimiento más crítico. Si un usuario reporta que sospecha que su clave privada ha sido expuesta:

1. **Aceptar el reporte sin pedir pruebas:** el coste de revocar erróneamente es bajo; el coste de no revocar y que sea real es altísimo.
2. **Revocar inmediatamente:**

- `/admin/certificates.php` → buscar serial → Revocar → motivo `keyCompromise`.

3. **Forzar CRL inmediata:** `/admin/crl.php → CA → Generar CRL`.
4. **Verificar propagación:**

```
# OCSP debe responder revoked en < 5 s
openssl ocsp -issuer /tmp/intermediate.crt -cert
/tmp/cert_comprometido.crt \
    -url http://ocsp.marchernandez.es/<slug> -resp_text -noverify |
grep "Cert Status"
# CRL debe contener el serial
curl -s http://crl.marchernandez.es/<slug>/ca.crl > /tmp/crl
openssl crl -in /tmp/crl -inform DER -text -noout | grep "Serial
Number"
```

5. **Notificar a las aplicaciones consumidoras** que cacheen OCSP que invaliden cache (en este sistema, no hay caching agresivo; en sistemas con OCSP stapling, reiniciar el web server).
6. **Auditar el periodo anterior:** revisar `sso_audit_log` y `audit_log` por uso de ese certificado en los últimos 30 días.
7. **Emitir certificado de reemplazo** al usuario (flujo normal de solicitud → aprobación → emisión).
8. **Documentar el incidente** en el registro interno (Anexo C.5).

C.5 Gestión de incidentes

Para cada tipo de incidente se documenta: **impacto · detección · respuesta · recuperación**.

C.5.1 Compromiso de clave de usuario

Campo	Contenido
Impacto	Bajo-medio. Limitado al perímetro del usuario. Posible suplantación hasta revocación.
Detección	Reporte del usuario / detección de uso anómalo (logs FNMT con horarios imposibles).

Campo	Contenido
Respuesta	Procedimiento Sección C.4.4 completo en menos de 15 minutos.
Recuperación	Emitir cert nuevo, comunicar al usuario, registrar incidente.

C.5.2 Compromiso de una CA intermedia

Campo	Contenido
Impacto	**Crítico.** Atacante puede firmar nuevos certificados aparentemente válidos.
Detección	Acceso no autorizado al servidor / passphrase comprometida / certificado firmado fuera de proceso.
Respuesta	Procedimiento Sección C.11.1 - revocación masiva de intermedia**.**
Recuperación	Crear nueva CA intermedia, re-emitir todos los certificados activos bajo la anterior.

C.5.3 Compromiso de la Root CA

Campo	Contenido
Impacto	Catastrófico**.** Toda la cadena de confianza es inválida.
Detección	Auditoría del servidor / passphrase robada / acceso físico al USB de backup.
Respuesta	(i) Comunicar a todos los relying parties que paren de confiar la Root; (ii) revocar todas las intermedias firmadas; (iii) re-keying total: generar nueva Root, nuevas intermedias, re-emitir todos los certs.

Campo	Contenido
Recuperación	Plan estimado: 2-4 semanas. Ver Sección C.11.2.

C.5.4 Pérdida de passphrase de una CA

Campo	Contenido
Impacto	Alto. La CA queda inutilizable (no se puede emitir ni firmar CRL).
Detección	Error "Error descifrando passphrase" en intentos de emisión.
Respuesta	**N**o es recuperable sin la passphrase. Aceptar la pérdida.
Recuperación	(i) Generar nueva CA del mismo nivel; (ii) re-emitir certificados activos; (iii) NO revocar los certs antiguos por `keyCompromise` (no está comprometida la clave, sólo es inaccesible) - usar `cessationOfOperation`; (iv) marcar la CA antigua como `is_active=0` pero conservar su CRL.

Mitigación preventiva: las passphrases deben estar en **dos** ubicaciones offline (un gestor de secretos en USB cifrado + sobre cerrado en caja fuerte). Nunca en un único punto.

C.5.5 Expiración accidental (cert o CRL)

Campo	Contenido
Impacto	Medio. Servicios que dependen del cert/CRL fallan hasta renovación.
Detección	Dashboard PKI marca expiraciones próximas. Apps consumidoras reportan errores TLS.

Campo	Contenido
Respuesta	Emitir cert nuevo (no se "renueva" técnicamente, se genera otro con misma CN).
Recuperación	El cert antiguo expirado no se revoca (ya no es válido por sí mismo).

C.5.6 CRL no regenerada

Campo	Contenido
Impacto	Alto. Clientes que descargan CRL recibirán una expirada y, según política, rechazarán todo.
Detección	Alerta visual en `/admin/crl.php` (CRL expirada en rojo). Alerta Telegram automática.
Respuesta	(i) `Admin → CRL → Generar CRL` manual; (ii) revisar `logs/crl_cron.log` para identificar causa; (iii) si el lock `crl_generate.lock` está huérfano, borrarlo y volver a probar.
Recuperación	Restaurar el cron systemd/cron del Anexo A Sección A.7.

C.5.7 Fallo del responder OCSP

Campo	Contenido
Impacto	Medio. Clientes con OCSP estricto rechazan certs; clientes con *soft-fail* siguen confiando.
Detección	`ocsp-health-check.sh` falla; usuarios reportan errores TLS.
Respuesta	Verificar `php-fpm` activo, BD accesible, certificado responder no expirado.

Campo	Contenido
Recuperación	`systemctl restart plesk-php84-fpm`. Si persiste, ver Sección C.10 hardening + logs Apache.

C.5.8 Abuso / bruteforce contra el SSO

Campo	Contenido
Impacto	Bajo (mitigado por rate limiting + TOTP) - medio si se combina con phishing.
Detección	`sso_audit_log` con muchos `login_failed` desde misma IP. `fail2ban` debería estar baneando.
Respuesta	(i) Verificar baneos `fail2ban-client status sso-api-bruteforce`; (ii) si la IP es persistente, bloqueo manual `firewall-cmd --add-rich-rule="rule source address=<IP> reject"`.
Recuperación	Bloqueo permanente, comunicación a usuario afectado para verificar si su cuenta sigue íntegra.

C.5.9 Fuga de clave de firma JWT

Campo	Contenido
Impacto	Crítico. Atacante puede emitir JWT válidos a cualquier identidad.
Detección	Acceso no autorizado a `sso/keys/*.pem`. Auditoría detecta `kid` desconocidos.
Respuesta	Procedimiento Sección C.8.1 - rotación de clave JWT con periodo cero de gracia (la nueva clave reemplaza a la comprometida sin overlap).

Campo	Contenido
Recuperación	Invalidar todas las sesiones (`mass_session_kill`). Forzar re-login de todos los usuarios.

C.6 Operación diaria

C.6.1 Checklist diario (10 min)

```
□ Abrir https://pki.marchernandez.es/admin/
  □ ¿Solicitudes pendientes? → procesar
  □ ¿Certificados expirando en <30 días? → avisar a los titulares
  □ ¿Alguna CRL expirada (rojo)? → escalado Sección C.5.6

□ Abrir https://sso.marchernandez.es/admin/sessions.php
  □ ¿Sesiones desde IPs inesperadas (países distintos)? → revisar

□ Revisar Telegram del bot operativo:
  □ Alertas OCSP / CRL / cron → triaje

□ Backups Plesk (panel Backup Manager):
  □ Último backup <24 h y verde
```

C.6.2 Checklist semanal (30 min)

```
□ Auditoría SSO:
  SELECT severity, COUNT(*) FROM sso_audit_log
  WHERE created_at > DATE_SUB(NOW(), INTERVAL 7 DAY)
  GROUP BY severity;
  □ ¿Picos inusuales de warning/critical? → investigar

□ Auditoría PKI:
  SELECT action, COUNT(*) FROM audit_log
  WHERE created_at > DATE_SUB(NOW(), INTERVAL 7 DAY)
  GROUP BY action ORDER BY 2 DESC;
  □ Comparar con la semana anterior (esperar variaciones <30%)

□ Revisar revocaciones de la semana:
  SELECT serial_number, common_name, revocation_reason, revocation_date
  FROM certificates
  WHERE status = 'revoked'
    AND revocation_date > DATE_SUB(NOW(), INTERVAL 7 DAY);
  □ ¿Algún keyCompromise? → revisar acción tomada

□ Estado de los timers:
  systemctl list-timers --all | grep -E "pki|sso"
  □ Todos en verde con LAST reciente

□ fail2ban activo:
  sudo fail2ban-client status
```

C.6.3 Checklist mensual (2 h)

```
□ Restauración de prueba (en VPS espejo, NO en producción):
  □ Restaurar backup del día anterior
  □ Verificar que el SSO levanta y permite login
  □ Verificar que la PKI emite un cert de prueba
  □ Documentar el tiempo total → debería ser <2 h (objetivo RNF-10)

□ Rotación de secretos (alternativo: trimestral):
  □ Regenerar client_secret de la app PKI (Admin → Aplicaciones → Regenerar
secret)
  □ Actualizar pki/config/app.php con el nuevo valor
  □ Reiniciar php-fpm
  □ Verificar login PKI vía SSO

□ Actualizaciones de sistema:
  sudo dnf check-update
  sudo dnf update -y       # ventana de mantenimiento
  sudo systemctl reboot    # si actualizó kernel

□ Actualización Plesk:
  Plesk → Herramientas → Actualizaciones → Aplicar parches de seguridad

□ Verificar SSL/TLS:
  curl -I https://sso.marchernandez.es/
  □ Caducidad LE > 14 días (debería renovar automáticamente)

□ Revisión de cuentas inactivas:
  SELECT username, last_login_at FROM sso_users
  WHERE last_login_at < DATE_SUB(NOW(), INTERVAL 90 DAY)
    AND is_active = 1;
  □ Desactivar las que aplique
```

C.7 Backup y recuperación

C.7.1 Tres capas de backup (no equivalentes)

Capa	Qué contiene	Frecuencia	Quién custodia	Cifrado adicional
(A) BD	Volcados SQL de ambas BDs	Diario	Plesk Backup Manager	Sí (passphrase Plesk)

Capa	Qué contiene	Frecuencia	Quién custodia	Cifrado adicional
(B) Claves CA	Ficheros `*.key` (cifrados con passphrase)	Diario + manual tras cambios	USB físico + Plesk	Doble (passphrase CA + USB)
(C) Passphrases	Passphrases en claro	Manual	Gestor de secretos offline	Maestra del gestor

Regla de oro: las tres capas deben estar en localizaciones físicamente distintas y nunca en la misma persona o dispositivo. Si las tres se pueden recuperar desde el mismo punto, el sistema no tiene backup real (tiene una redundancia única).

C.7.2 Backup automático (Plesk)

Plesk → Herramientas → Backup Manager:

```
Frecuencia: Diaria a las 04:00 UTC
Contenido: Archivos · BDs · Configuración Plesk
Retención local: 7 días
Retención remota: 30 días (en S3/B2 vía Plesk Extension)
Cifrado: Activado con passphrase
Compresión: gzip
Notificación email al fallar: Sí
```

C.7.3 Backup manual de BDs (puntual)

```
# Backup completo con timestamp
TS=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR=/var/backups/pki_sso
sudo mkdir -p $BACKUP_DIR

# SSO
mysqldump --single-transaction --quick --routines --triggers \
    -u admin_sso_nuevo_2 -p marchernandezmo_sso_nuevo_2 \
    | gzip > $BACKUP_DIR/sso_${TS}.sql.gz

# PKI
mysqldump --single-transaction --quick --routines --triggers \
    -u admin_pki_nuevo -p marchernandezmo_pki_nuevo \
    | gzip > $BACKUP_DIR/pki_${TS}.sql.gz

# Permisos
sudo chmod 600 $BACKUP_DIR/*.gz
sudo chown root:root $BACKUP_DIR/*.gz
```

C.7.4 Backup de claves de CA (cifrado adicional)

```
# Empaquetar y cifrar con OpenSSL antes de subir a cualquier sitio
TS=$(date +%Y%m%d_%H%M%S)
sudo tar -czf - -C /var/www/vhosts/marchernandez.es/pki.marchernandez.es ca/
\
    | openssl enc -aes-256-cbc -salt -pbkdf2 \
        -out /var/backups/pki_sso/ca_keys_${TS}.tar.gz.enc

# Permisos restrictivos
sudo chmod 400 /var/backups/pki_sso/ca_keys_${TS}.tar.gz.enc
```

La passphrase de este cifrado no es la de las CAs. Es una passphrase de transporte independiente, guardada en el gestor de secretos offline.

C.7.5 Backup de passphrases (gestor offline)

Tras cualquier cambio de passphrase (creación de CA, rotación, etc.):

1. Anotar en el gestor de secretos (KeePassXC, recomendado por ser local y cifrado).
2. Exportar el `.kdbx` a un USB cifrado.
3. Guardar el USB en una caja fuerte física o sobre cerrado con sellado.
4. Mantener dos copias en ubicaciones distintas.
5. Probar la legibilidad del USB cada 6 meses (los flash pierden carga).

C.7.6 Restauración completa (desastre total)

Estimación: 2-4 h en hardware similar.

```
Paso 0:  Preparar VPS nuevo con AlmaLinux 9 + Plesk (≈30 min)
Paso 1:  Restaurar último backup Plesk (archivos + BDs + config)
Paso 2:  Restaurar claves de CA desde USB cifrado:
         openssl enc -d -aes-256-cbc -pbkdf2 \
             -in ca_keys_*.tar.gz.enc | tar -xzf - -C /var/www/vhosts/...
Paso 3:  Verificar que las passphrases del gestor offline siguen funcionando:
         openssl rsa -in ca/root/root_ca.key -noout -check
Paso 4:  Apuntar DNS a la nueva IP del VPS
Paso 5:  Ejecutar smoke test del Anexo A Sección A.9 completo
Paso 6:  Notificar a usuarios la ventana de mantenimiento finalizada
```

C.7.7 Restauración parcial (un cert borrado por error, una CA mal configurada)

Caso	Procedimiento
Borraron una fila de `certificates`	Restaurar sólo esa fila desde backup SQL: `mysqlbinlog` o `mysqldump --where`
Una CA tiene `index.txt` corrupto	Reconstruir desde `certificates` table: SQL → archivo `index.txt`
Plantilla mal configurada emitió certs erróneos	Revocar los certs (`superseded`), corregir plantilla, re-emitir
Falló una rotación JWT (ningún `kid` activo)	Restaurar `sso_keypairs` fila por fila + archivo `.pem` correspondiente

C.8 Rotación de claves

C.8.1 Rotación de la clave de firma JWT (RS256)

Calendario recomendado: cada 180 días (semestralmente) o inmediatamente ante compromiso.

Procedimiento con periodo de gracia (rotación normal):

```
cd /var/www/vhosts/marchernandez.es/sso.marchernandez.es/

# 1. Generar nuevo keypair
NEW_KID=$(openssl rand -hex 16)
openssl genrsa -out keys/${NEW_KID}.pem 2048
chmod 600 keys/${NEW_KID}.pem
chown marchernandez:psacln keys/${NEW_KID}.pem

# 2. Extraer la pública
PUB=$(openssl rsa -in keys/${NEW_KID}.pem -pubout 2>/dev/null)

# 3. Insertar en BD con la nueva clave como activa, la anterior aún activa
mysql -u admin_sso_nuevo_2 -p marchernandezmo_sso_nuevo_2 <<EOF
INSERT INTO sso_keypairs (kid, key_file_path, public_key_pem, algorithm,
is_active)
VALUES ('${NEW_KID}', '${NEW_KID}.pem', "${PUB}", 'RS256', 1);
EOF

# En este momento JWKS publica ambas claves (kid antiguo + kid nuevo).
# El SSO firma nuevos JWTs con la última (la recién creada).
# Los JWTs antiguos siguen validándose con la clave antigua.
```

Tras el periodo de gracia (24 h, suficiente porque los JWT caducan en 1 h):

```
-- 4. Desactivar la clave antigua (deja de validar JWTs nuevos pero el cron
cleanup
--    la borrará tras 30 días si is_active=0)
UPDATE sso_keypairs SET is_active = 0
WHERE kid != '${NEW_KID}' AND is_active = 1;
```

Rotación de emergencia (compromiso, periodo cero):

```
-- Desactivar inmediatamente todas las claves antiguas
UPDATE sso_keypairs SET is_active = 0;

-- Forzar logout de todos los usuarios (sus JWTs ya no validan)
UPDATE sso_sessions SET is_active = 0 WHERE is_active = 1;
```

Y proceder con la generación de la nueva clave igual que el procedimiento normal, pero sin esperar el periodo de gracia.

C.8.2 Rotación de CA intermedia

Cuándo: validez restante < 3 años, compromiso, o cambio de política (p. ej. migrar a ECDSA en el futuro).

Procedimiento con solapamiento (no traumático):

1. Crear nueva CA intermedia con CN distinto: "Omnipresence TrustCA Intermediate 2027".

2. Las dos CAs coexisten: la antigua sigue emitiendo (o ya sólo firma CRLs) y la nueva empieza a emitir.
3. Durante el periodo de transición (3-6 meses), las solicitudes nuevas usan la nueva CA.
4. Cuando la CA antigua queda sin certs activos, marcarla `is_active=0` (no se borra: sigue sirviendo CRL de los revocados históricos).
5. Tras la caducidad natural del último cert emitido, retirar la CA antigua del repositorio AIA.

C.8.3 Publicación simultánea de varios `kid`

El endpoint `/api/jwks.php` ya soporta múltiples claves: devuelve todas las filas con `is_active = 1` en el array `keys[]`. Los clientes que validan JWTs deben respetar el campo `kid` del header y buscar la clave correspondiente.

Verificar:

```
curl -s https://sso.marchernandez.es/api/jwks.php | jq '.keys | length'
# Durante rotación: 2. En estado normal: 1.
```

C.9 Logs y auditoría

C.9.1 Inventario de logs

Fuente	Ubicación	Retención
`sso_audit_log` (BD)	Tabla `marchernandezmo_sso_nuevo_2.sso_audit_log`	90 d / críticos perpetuo
`audit_log` PKI (BD)	Tabla `marchernandezmo_pki_nuevo.audit_log`	Perpetuo
`ocsp_queries` (BD)	Tabla PKI; volumen alto, considerar particionado mensual a futuro	30 d

Fuente	Ubicación	Retención
`crl_downloads` (BD)	Tabla PKI	90 d
`sso_login_logs` (BD)	Tabla SSO; logins exitosos y fallidos	180 d
PHP errores SSO	`/var/www/vhosts/marchernandez.es/logs/sso.marchernandez.es_php_error.log`	Rotación Plesk
PHP errores PKI	`/var/www/vhosts/marchernandez.es/logs/pki.marchernandez.es_php_error.log`	Rotación Plesk
Apache access/error	`/var/log/httpd/*.log` + `/var/www/vhosts/system/<dominio>/logs/`	Rotación logrotate
Cron CRL	`/var/www/vhosts/marchernandez.es/pki.marchernandez.es/logs/crl_cron.log`	Rotación manual
Cron SSO cleanup	`/var/www/vhosts/marchernandez.es/sso.marchernandez.es/logs/cron.log`	Rotación manual
fail2ban	`/var/log/fail2ban.log`	Rotación logrotate

C.9.2 Consultas SQL útiles

SSO - Top usuarios con más fallos de login últimas 24 h:

```sql
SELECT JSON_EXTRACT(details, '$.username') AS user_attempted,
       COUNT(*) AS attempts,
       MAX(created_at) AS last_attempt
FROM sso_audit_log
WHERE event_type = 'login_failed'
  AND created_at > DATE_SUB(NOW(), INTERVAL 24 HOUR)
GROUP BY user_attempted
ORDER BY attempts DESC
LIMIT 20;
```

SSO - Detectar reutilización de sesión desde múltiples IPs (posible robo):

```sql
SELECT user_id, COUNT(DISTINCT ip_address) AS distinct_ips,
       GROUP_CONCAT(DISTINCT ip_address) AS ips
FROM sso_sessions
WHERE is_active = 1
  AND created_at > DATE_SUB(NOW(), INTERVAL 7 DAY)
GROUP BY user_id
HAVING distinct_ips > 3;
```

PKI - Certificados emitidos por usuario en el último mes:

```sql
SELECT u.username, COUNT(*) AS certs_emitted
FROM certificates c
JOIN pki_users u ON u.id = c.issued_by
WHERE c.created_at > DATE_SUB(NOW(), INTERVAL 30 DAY)
GROUP BY u.username
ORDER BY certs_emitted DESC;
```

PKI - Revocaciones por motivo (estadística operacional):

```sql
SELECT revocation_reason, COUNT(*) AS total
FROM certificates
WHERE status = 'revoked'
GROUP BY revocation_reason
ORDER BY total DESC;
```

PKI - Volumen de consultas OCSP por hora (detectar abuso):

```sql
SELECT DATE_FORMAT(created_at, '%Y-%m-%d %H:00') AS hour,
       COUNT(*) AS queries
FROM ocsp_queries
WHERE created_at > DATE_SUB(NOW(), INTERVAL 48 HOUR)
GROUP BY hour
ORDER BY hour;
```

Cross - Certificados expirando en <30 días:

```
SELECT serial_number, common_name, issued_to_email,
       valid_to,
       DATEDIFF(valid_to, NOW()) AS days_remaining
FROM certificates
WHERE status = 'valid'
  AND valid_to BETWEEN NOW() AND DATE_ADD(NOW(), INTERVAL 30 DAY)
ORDER BY valid_to ASC;
```

C.9.3 Filtros Apache rápidos

```
# Top 10 IPs por hits en el SSO en las últimas 24h
sudo grep "$(date -d '24 hours ago' '+%d/%b/%Y')"
/var/www/vhosts/system/sso.marchernandez.es/logs/access_ssl_log \
  | awk '{print $1}' | sort | uniq -c | sort -rn | head -10

# Errores 5xx en el PKI hoy
sudo grep "$(date '+%d/%b/%Y')"
/var/www/vhosts/system/pki.marchernandez.es/logs/access_ssl_log \
  | awk '$9 >= 500 {print}' | tail -50

# Intentos de SQL injection (detectar payloads en logs)
sudo grep -iE "(union.*select|';|--%20|<script)"
/var/www/vhosts/system/sso.marchernandez.es/logs/access_ssl_log
```

C.9.4 fail2ban

```
# Ver estado general
sudo fail2ban-client status

# Ver estado de un jail concreto
sudo fail2ban-client status sso-api-bruteforce

# Desbanear una IP (típico: usuario legítimo baneado por error)
sudo fail2ban-client set sso-api-bruteforce unbanip 1.2.3.4

# Banear manualmente
sudo fail2ban-client set sso-api-bruteforce banip 1.2.3.4
```

C.10 Hardening y buenas prácticas

Práctica	Estado actual	Acción recomendada
Root CA realmente offline	Parcial. La clave existe en el	A futuro, mover a HSM o servidor air-gapped. Documentado como limitación en Sección 9.3.

Práctica	Estado actual	Acción recomendada
	servidor pero solo se descifra a puntualmente.	
MFA obligatorio para administradores	Implementado	Verificar `totp_enabled=1` para todos los admins:
		`SELECT username FROM sso_users WHERE role IN ('admin','superadmin') AND totp_enabled=0;`
Principio del menor privilegio	Parcial	Plesk crea usuarios MariaDB con `ALL PRIVILEGES`; revocar y conceder sólo `SELECT, INSERT, UPDATE, DELETE` (Anexo A Sección A.4.2).
Separación de funciones	Implementado	`superadmin` ≠ `admin` ≠ `operator`. Mantener al menos un usuario por rol.
Backups offline	Obligatorio	Las claves CA fuera de Plesk y del propio servidor.
No reutilizar passphrases	Manual	Auditoría visual del gestor de secretos: cada CA debe tener una passphrase única.
MariaDB no expuesta al exterior	Sí	Verificar mensualmente con: `ss -tlnp | grep 3306` (sin resultado).
Revisar certificados próximos a expirar	Dashboard	Checklist semanal Sección C.6.2.

Práctica	Estado actual	Acción recomendada
Sesiones admin con timeout corto	Sí	JWT 1 h, refresh 24 h. Admins en máquinas no compartidas.
Logs íntegros (append-only)	Parcial	Hoy es teórico; en producción real, replicar logs a un servidor externo.
Actualizaciones de sistema mensuales	Manual	Calendario fijado Sección C.6.3.
Cifrado en reposo	Parcial	Claves CA cifradas; backups cifrados; BD no cifrada en disco (depende del filesystem).
Snapshots de VPS	Recomendado	Activar snapshots semanales del proveedor cloud (Ionos en el caso de referencia).

C.11 Procedimientos de emergencia

C.11.1 Revocación masiva de una CA intermedia

Situación: se sospecha o confirma que una CA intermedia ha sido comprometida.

```
Tiempo total estimado: 30-60 min (sin contar la re-emisión posterior)
```

Pasos:

1. **Aislar la CA comprometida:**

   ```
   UPDATE certificate_authorities SET is_active = 0
   WHERE id = <id_ca_comprometida>;
   ```

   Esto impide cualquier nueva emisión bajo esa CA.

2. **Revocar el certificado de la propia CA intermedia** (firmado por la Root):

- `/admin/certificates.php` → buscar el cert de la intermedia → Revocar → motivo `cACompromise`.

3. **Marcar todos los certificados emitidos por esa CA como revocados:**

```
UPDATE certificates
SET status = 'revoked',
    revocation_reason = 'cACompromise',
    revocation_date = NOW(),
    revoked_by = <id_admin>
WHERE ca_id = <id_ca_comprometida> AND status = 'valid';
```

4. **Regenerar la CRL de la Root** (que ahora incluye el cert de la intermedia revocada):
   - `/admin/crl.php → Root CA → Generar CRL`.
5. **Regenerar la CRL de la intermedia comprometida** una última vez (todos los certs ahora aparecen revocados):
   - `/admin/crl.php → CA intermedia → Generar CRL`.
6. **Comunicar a los relying parties:** email + Telegram + cartel en `https://ca.marchernandez.es` explicando la situación.
7. **Crear una nueva CA intermedia** (Sección C.3.2) y **re-emitir** los certificados que estaban activos.
8. **Registrar el incidente** en bitácora con: fecha de detección, causa, alcance, certificados afectados.

C.11.2 Rollback tras una restauración fallida

Situación: una restauración (de backup) ha dejado el sistema en estado inconsistente: BDs vs. ficheros desincronizados, CRLs con seriales que ya no existen, etc.

Pasos:

1. **Activar modo mantenimiento** (Sección C.11.3) para evitar que usuarios encuentren errores.

2. **Identificar el punto consistente más reciente** (combinación de backup BD + backup ficheros del mismo día).
3. **Restaurar BD y ficheros en pareja** (no mezclar fechas).
4. **Verificar consistencia:**

```
-- ¿Todas las CAs activas tienen su clave privada en disco?
SELECT id, name FROM certificate_authorities WHERE is_active = 1;
```

   Comprobar a mano que `pki/ca/<slug>/intermediate_ca.key` existe para cada una.

5. **Regenerar CRLs frescas** para todas las CAs activas.
6. **Smoke test completo** (Anexo A Sección A.9).
7. **Desactivar modo mantenimiento.**

C.11.3 Deshabilitar temporalmente la emisión de certificados

Situación: hay que parar todas las emisiones (incidente, auditoría externa, ventana de mantenimiento de la Root).

```
-- Marcar todas las CAs como inactivas (no emiten, pero siguen sirviendo
CRL/OCSP)
UPDATE certificate_authorities SET is_active = 0;
```

UI también muestra mensaje "CA no disponible para emisión" en `/request.php`. Para reactivar:

```
UPDATE certificate_authorities SET is_active = 1 WHERE name LIKE
'Omnipresence%';
-- o solo las que se quieran reactivar
```

C.11.4 Desactivar el login con FNMT/DNIe

Situación: se detecta un fallo de validación de certificados FNMT (p. ej. una CA intermedia FNMT ha cambiado y aún no se ha actualizado en `trusted_ca_bundle.pem`).

```
-- Desactivar trusted issuers FNMT/DNIe en SSO
UPDATE sso_trusted_issuers SET is_active = 0
WHERE description LIKE '%FNMT%' OR description LIKE '%DNI%';
```

Los usuarios que sólo usaran login por certificado podrán seguir entrando con su contraseña + TOTP (deberían tener al menos uno de los dos métodos configurados).

Para reactivar tras corregir `trusted_ca_bundle.pem`:

```
UPDATE sso_trusted_issuers SET is_active = 1 WHERE issuer_type IN
('certificate','smartcard');
```

C.11.5 Modo mantenimiento (todos los servicios)

```
# Activar página estática "En mantenimiento" en SSO y PKI
sudo cp
/var/www/vhosts/marchernandez.es/sso.marchernandez.es/public/.htaccess \

/var/www/vhosts/marchernandez.es/sso.marchernandez.es/public/.htaccess.bak

sudo tee
/var/www/vhosts/marchernandez.es/sso.marchernandez.es/public/.htaccess
<<'EOF'
RewriteEngine On
RewriteCond %{REMOTE_ADDR} !^TU.IP.DE.ADMIN$
RewriteCond %{REQUEST_URI} !^/maintenance.html$
RewriteCond %{REQUEST_URI} !\.(css|js|png|svg|woff2?)$
RewriteRule ^ /maintenance.html [R=503,L]
ErrorDocument 503 /maintenance.html
EOF

# (Crear /maintenance.html con mensaje claro)

# Repetir para pki.marchernandez.es

# Para salir del modo mantenimiento:
sudo mv
/var/www/vhosts/marchernandez.es/sso.marchernandez.es/public/.htaccess.bak \

/var/www/vhosts/marchernandez.es/sso.marchernandez.es/public/.htaccess
```

C.12 Multiorganización

Esta sección desarrolla el detalle operativo del modo multiorganización descrito en el apartado 7.4.6 del TFG. La PKI admite hasta cuatro escenarios distintos sobre la misma instalación sin cambios de código, gracias a la separación entre tres conceptos (CA emisora, plantilla de certificado y solicitud) y, en particular, a la columna `ca_id` opcional de la tabla `certificate_templates`.

C.12.1 Concepto y pieza clave de configuración

La clave del diseño es el comportamiento de la columna `ca_id` de `certificate_templates`:

- Si `ca_id` está rellena con el `id` de una CA, la plantilla sólo puede emitirse desde esa CA.
- Si `ca_id` es `NULL`, la plantilla es posible emitirla desde cualquier CA activa.

Esta semántica simple genera los cuatro escenarios operativos. La elección del escenario es una decisión del administrador: la base de datos no impone ninguno por defecto más allá del estado inicial tras la instalación.

Escenario	`ca_id` en plantillas	Casos de uso
Despliegue unifamiliar	`NULL` en todas las plantillas, una sola intermedia activa	Organización pequeña con una única familia de certificados
Segregación por uso	`ca_id` rellena, una intermedia por familia de uso	Separar TLS de servidor, S/MIME, código y VPN en CAs distintas
Multiorganización	`ca_id` rellena, una intermedia por organización cliente	Proveedor de servicios que emite para varias entidades
Sandbox de pruebas	CA intermedia adicional con `is_active=0` salvo durante pruebas	Validar plantillas y procedimientos sin tocar producción

Tabla C.12.1. Cuatro escenarios soportados por la misma instalación.

C.12.2 Escenario 1: Despliegue unifamiliar

Es el escenario por defecto tras la instalación (descrita en el Anexo A) y el activo en el despliegue de referencia de marchernandez.es.

Características:

- Una única CA intermedia activa que firma todos los perfiles.
- Todas las plantillas con `ca_id = NULL`.
- No hay segregación entre familias de uso: el administrador valora cada solicitud por su contenido.

Configuración (consulta de verificación):

```
SELECT id, slug, display_name, ca_id
FROM certificate_templates
WHERE is_active = 1;
```

Resultado esperado: las seis plantillas activas, todas con `ca_id` a `NULL`.

```
SELECT id, type, subject_dn, is_active
FROM certificate_authorities
WHERE is_active = 1
ORDER BY type;
```

Resultado esperado: 1 raíz y 1 (o N) intermedia/s activas.

C.12.3 Escenario 2: Segregación por uso

Aplica cuando la organización quiere que cada familia de uso (TLS de servidor, S/MIME, firma de código, etc.) cuelgue de una CA intermedia distinta. Esto facilita la rotación independiente y el control de impacto: si se compromete la intermedia de firma de código, los certificados TLS de servidor no se ven afectados.

Procedimiento de configuración:

1. Crear las CAs intermedias adicionales (procedimiento documentado en la sección C.3.4 sobre creación de CAs).
2. Vincular cada plantilla a su CA correspondiente:

```
UPDATE certificate_templates
SET ca_id = (SELECT id FROM certificate_authorities WHERE subject_dn =
'CN=Intermediate TLS, O=Mi Org, C=ES')
WHERE slug IN ('server-auth');

UPDATE certificate_templates
SET ca_id = (SELECT id FROM certificate_authorities WHERE subject_dn =
'CN=Intermediate SMIME, O=Mi Org, C=ES')
WHERE slug IN ('smime-email');

UPDATE certificate_templates
SET ca_id = (SELECT id FROM certificate_authorities WHERE subject_dn =
'CN=Intermediate Code, O=Mi Org, C=ES')
WHERE slug IN ('code-signing');
```

3. Verificación:

```
SELECT t.slug, t.display_name, c.subject_dn AS issuer
FROM certificate_templates t
JOIN certificate_authorities c ON c.id = t.ca_id
WHERE t.is_active = 1
ORDER BY t.slug;
```

Aviso. Las solicitudes que el portal acepte después de este cambio quedarán vinculadas a la CA configurada. Las solicitudes en curso (`status = 'pending'`) lanzadas antes del cambio mantienen el `ca_id` original; revíselas antes de la transición.

C.12.4 Escenario 3: Multiorganización

Aplica cuando la misma instalación da servicio a varias organizaciones que deben permanecer aisladas. Cada organización cliente recibe una CA intermedia con su nombre, y sus plantillas se vinculan exclusivamente a esa CA.

Procedimiento de alta de una organización cliente:

1. Crear la CA intermedia para la organización:

```
INSERT INTO certificate_authorities (parent_ca_id, type, subject_dn,
key_file_path, passphrase_file_path, is_active)
VALUES (
  (SELECT id FROM certificate_authorities WHERE type = 'root' AND is_active =
1),
  'intermediate',
  'CN=Intermediate Acme, O=Acme S.A., C=ES',
  '/var/www/vhosts/.../pki/ca/intermediate_acme_ca.pem',
  '/var/www/vhosts/.../pki/ca/intermediate_acme_passphrase.enc',
  1
);
```

2. Replicar las plantillas necesarias (o crear nuevas) vinculadas a esa CA:

```
INSERT INTO certificate_templates (slug, display_name, openssl_cnf_path,
ca_id, validity_days, is_active)
SELECT
  CONCAT(slug, '-acme'),
  CONCAT(display_name, ' [Acme]'),
  openssl_cnf_path,
  (SELECT id FROM certificate_authorities WHERE subject_dn = 'CN=Intermediate
Acme, O=Acme S.A., C=ES'),
  validity_days,
  1
FROM certificate_templates
WHERE ca_id IS NULL
  AND slug IN ('server-auth', 'client-auth', 'smime-email');
```

3. Restringir el acceso a las plantillas de Acme mediante `sso_applications` (si se utiliza alguna aplicación interna del portal PKI con `access_mode='restricted'`).

4. Verificación de aislamiento:

```
SELECT t.slug, t.display_name, c.subject_dn
FROM certificate_templates t
JOIN certificate_authorities c ON c.id = t.ca_id
WHERE c.subject_dn LIKE '%Acme%'
  AND t.is_active = 1;
```

Aviso. En este escenario el panel de administración debe restringirse por organización: un administrador de Acme no debe ver las solicitudes de otra organización. La forma recomendada es habilitar el filtro por `ca_id` en `pki/public/admin/requests.php` (ya soportado por el código) y asociar cada cuenta de administrador a su CA mediante un campo adicional `admin_ca_scope` en `pki_users` o el sistema RBAC equivalente.

C.12.5 Escenario 4: Sandbox de pruebas

Aplica para validar nuevas plantillas, perfiles X.509 o procedimientos sin riesgo de afectar a los certificados emitidos en producción.

Procedimiento:

1. Crear una CA intermedia adicional marcada como inactiva:

```
INSERT INTO certificate_authorities (parent_ca_id, type, subject_dn,
key_file_path, passphrase_file_path, is_active)
VALUES (
  (SELECT id FROM certificate_authorities WHERE type = 'root' AND is_active =
1),
  'intermediate',
  'CN=Intermediate Sandbox, O=Mi Org, C=ES',
  '/var/www/vhosts/.../pki/ca/intermediate_sandbox_ca.pem',
  '/var/www/vhosts/.../pki/ca/intermediate_sandbox_passphrase.enc',
  0
);
```

2. Activar la CA sólo durante la ventana de pruebas:

```
UPDATE certificate_authorities SET is_active = 1
WHERE subject_dn = 'CN=Intermediate Sandbox, O=Mi Org, C=ES';
```

3. Vincular las plantillas en pruebas a la CA *sandbox*:

```
UPDATE certificate_templates SET ca_id = (
  SELECT id FROM certificate_authorities WHERE subject_dn = 'CN=Intermediate
Sandbox, O=Mi Org, C=ES'
)
WHERE slug = 'new-experimental-profile';
```

4. Al cerrar la ventana, desactivar la CA y restablecer `ca_id` a `NULL` o a la CA destino:

```
UPDATE certificate_authorities SET is_active = 0
WHERE subject_dn = 'CN=Intermediate Sandbox, O=Mi Org, C=ES';
```

Aviso. Los certificados emitidos desde la CA *sandbox* no son confiables fuera del entorno de pruebas. Asegúrese de revocarlos al cerrar la ventana o de excluirlos del repositorio público de CRL/OCSP si no quiere arrastrar ruido en las métricas de validación.

C.12.6 Runbook de migración entre escenarios

La transición entre escenarios sin pérdida de servicio sigue cinco pasos. Tome este *runbook* como referencia general y adáptelo a la combinación origen y destino concreto.

Paso	Acción	Verificación
1	Programar ventana de mantenimiento y avisar a los usuarios.	Aviso publicado en el portal y enviado por Telegram.
2	Realizar copia de seguridad completa (BD + `pki/ca/`).	`pki-backup.timer` ejecutado manualmente con éxito.
3	Crear las nuevas CAs intermedias requeridas (Escenario destino).	`SELECT ... FROM certificate_authorities WHERE is_active = 1;` devuelve las CAs esperadas.
4	Actualizar `ca_id` en `certificate_templates` para reflejar la nueva vinculación.	Consultas SQL de verificación de cada escenario (apartados C.12.2 a C.12.5).
5	Forzar la regeneración de CRL y respuesta OCSP para todas las CAs activas.	`pki-crl-rotate.timer` ejecutado manualmente; OCSP responde a una consulta de prueba.

Tabla C.12.2. Runbook genérico de migración entre escenarios multiorganización.

Aviso final. El cambio de escenario no afecta a los certificados ya emitidos: cada certificado mantiene su cadena hasta su `notAfter`. La separación entre escenarios se aplica a las nuevas solicitudes desde el momento de la transición.

C.13 Referencias internas

C.13.1 Cruces a otros anexos

Tema	Anexo / sección
Instalación inicial del sistema	Anexo A
Plantillas OpenSSL (`.cnf`)	Anexo F (Sección F.2 Root, Sección F.3 perfiles)
Esquema completo de BD	Anexo E (Sección E.2 SSO, Sección E.3 PKI, Sección E.5 migraciones)
Endpoints y API	Anexo D (Sección D.3 SSO, Sección D.4 PKI, Sección D.5 públicos)
Configuración Apache / cabeceras HTTP	Manual de Código Relevante (Sección A.7) (fragmentos relevantes Apache + CSP)
Código relevante seleccionado	Manual de Código Relevante (catálogo completo)

C.13.2 Cruces al cuerpo del TFG

Tema	Sección del cuerpo
Jerarquía PKI (raíz + intermedias)	Sección 7.4 (especialmente Sección 7.4.1, Sección 7.4.3)
Almacenamiento de claves CA y passphrases	Sección 7.4.4
Revocación, CRL y OCSP	Sección 7.4.5, Sección 7.7, Sección 7.8
Métodos de autenticación SSO	Sección 7.5, Sección 7.6
OAuth 2.0 + PKCE + OIDC	Sección 7.5.1

Tema	Sección del cuerpo
Binding de sesión (UA + IP) como señal de riesgo	Sección 7.5.3
Seguridad transversal (las 9 capas + STRIDE)	Sección 7.8, Sección 7.9
Despliegue y operación	Sección 7.10
Limitaciones reconocidas (Root no offline, etc.)	Sección 9.3
Vías futuras (HSM, OCSP responder dedicado...)	Sección 9.4

C.13.3 Glosario rápido para incidentes

- **AIA** (Authority Information Access): extensión X.509 con URL del cert de la CA emisora. En este sistema: `http://ca.marchernandez.es/...`.
- **CDP** (CRL Distribution Points): URL de la CRL. `http://crl.marchernandez.es/...`.
- `kid`: identificador de la clave de firma en el header del JWT y en la entrada del JWKS.
- **OCSP** (Online Certificate Status Protocol): consulta puntual sobre revocación. Más fresco que CRL.
- **Passphrase de CA**: contraseña que cifra la clave privada PEM con AES-256-CBC.
- `access_mode = restricted`: una aplicación cliente OAuth sólo admite usuarios en su allowlist.

Para definiciones completas, ver el glosario al inicio del TFG.

Última nota operativa. Este manual es un documento vivo. Cada incidente real debe quedar reflejado aquí (en un appendix interno, no necesariamente

publicado) con: fecha, severidad, causa raíz, acciones tomadas y lecciones aprendidas. La diferencia entre un sistema bien administrado y uno frágil suele ser la disciplina de mantener actualizado este tipo de manual, no la sofisticación técnica.