

FASE II: AUTH CON RBAC — ISS-14

Feature RefreshTokens: Sesiones Renovables, Opacos y Revocables

Gestión de sesiones de larga duración con rotación estricta de tokens opacos (SHA-256), revocación en BD y soporte multi-dispositivo.

Propósito y Arquitectura de Sesión Renovables

Access Token (JWT)

Corta Duración (15m - 1h)

Token JWT firmado con HS256. Sin estado (stateless) para rápida verificación en cada petición mediante el middleware authenticate.

Refresh Token (Opaco)

Larga Duración (7d - 30d)

Cadena aleatoria criptográfica segura de 64 caracteres. Con estado (stateful), persistido como Hash SHA-256 en la tabla refresh_tokens.

Rotación Estricta

Invalidación Inmediata

Al refrescar la sesión, el refresh token usado se marca como revocado (revoked_at = NOW()) y se emite un nuevo par de tokens.

Anatomía de la Tabla refresh_tokens y SHA-256

ESTRUCTURA EN BASE DE DATOS

- id (INTEGER / BIGINT): Clave primaria autoincremental
- user_id (FK -> users.id): Identifica al propietario del token
- token_hash (VARCHAR(64)): Hash SHA-256 del token opaco (NUNCA el token crudo)
- expires_at (DATETIME): Fecha límite de validez del token
- revoked_at (DATETIME NULL): Timestamp de revocación (NULL = activo)
- ip_address / user_agent (VARCHAR): Auditoria y contexto del cliente

PRINCIPIOS DE SEGURIDAD CRÍTICOS

Tokens Opacos

Generados mediante
`crypto.randomBytes(32).toString('hex')`.
Impredecibles de 64 caracteres.

Almacenamiento Seguro

Si la base de datos se compromete, los atacantes solo obtienen hashes SHA-256 inservibles.

Detección de Reuso

Un intento de usar un token ya revocado (`revoked_at != NULL`) dispara una alerta y revoca todas las sesiones del usuario.

Flujo de Rotación Estricta (RefreshToken Rotation)

1. Petición Refresh

Cliente envía refresh_token crudo a POST /api/auth/refresh

2. Hash & Lookup

Servidor calcula SHA-256(refresh_token) y busca en BD

3. Validaciones

Verifica expires_at > NOW() y revoked_at IS NULL

4. Revocación

Marca el token actual como revocado (revoked_at = NOW())

5. Emisión Par

Genera nuevo Access Token (JWT) y nuevo Refresh Token

6. Respuesta Clean

Retorna el nuevo par de tokens al cliente en JSON

Endpoints de Revocación: Local y Global

POST /api/auth/refresh — Renovación de Sesión

Invalida el token actual y emite un nuevo par. Requiere { refresh_token } en el body.

POST /api/auth/revoke — Logout Local (Revocación Individual)

Invalida unicamente el refresh_token enviado. Ideal para cerrar sesión en el dispositivo actual.

POST /api/auth/revoke-all — Logout Global (Revocación Masiva)

Invalida TODOS los refresh tokens activos del usuario (revoked_at = NOW()). Fuerza la salida en todos los dispositivos.

Arquitectura del Feature: Service & Repository

RefreshTokensRepository

- `create(data)`: Persiste un nuevo hash de refresh token
- `findByHash(tokenHash)`: Busca por el hash SHA-256 calculado
- `revoke(id)`: Asigna `revoked_at = NOW()` a un token específico
- `revokeAllForUser(userId)`: Revoca masivamente todos los tokens del usuario
- `deleteExpired()`: Limpieza de mantenimiento de tokens caducados

RefreshTokensService

- `refresh(refreshTokenRaw)`: Valida estado, revoca actual, emite nuevo par
- `revokeToken(refreshTokenRaw)`: Valida y marca como revocado el token dado
- `revokeAllUserTokens(userId)`: Ejecuta revocación masiva y limpia sesiones
- `AppError Exception Handling`: Lanza 401 Unauthorized para tokens expirados o revocados

Especificación OpenAPI / Swagger: Auth / Sesiones

REGISTRO DE ENDPOINTS BAJO LA ETIQUETA 'Auth / Sesiones'

- `auth.swagger.ts`: Centraliza las especificaciones Swagger de renovación y revocación de tokens.
- Respuestas Estandarizadas: HTTP 200 OK con el par de tokens renovados; HTTP 401 Unauthorized ante tokens inválidos, expirados o ya revocados.
- Schemas DTO Integrados: `RefreshTokenInputDto`, `TokenResponseDto` y `MessageResponseDto` registrados en el componente OpenAPI global.
- Exposición UI: Disponibles para prueba interactiva en <http://localhost:4000/api/docs>.

Criterios de Cierre GATE (DoD del ISS-14)

CHECKLIST OBLIGATORIO DE VERIFICACIÓN (GATE VERDE)

- [✓] 1. **Compilación Estricta:** `npx tsc --noEmit` ejecuta con 0 errores de tipos.
- [✓] 2. **Creación de Tabla:** Tabla `refresh_tokens` creada y relacionada mediante FK con `users`.
- [✓] 3. **Rotación Funcional:** `POST /api/auth/refresh` invalida el token usado y emite uno nuevo.
- [✓] 4. **Detección de Reuso:** Intentar reusar un token revocado o expirado devuelve `401 Unauthorized`.
- [✓] 5. **Revocación Global:** `POST /api/auth/revoke-all` invalida adecuadamente todas las sesiones del usuario.
- [✓] 6. **Habilitación del ISS-15:** Queda lista la plataforma para el Cierre de Auth y login/sesión completa.