# MÓDULO DE SEGURIDAD – ASSIP‑ERP

Empresa: ERP ASSIP SOLIDARIA Y FINANCIERA

Autor: Carlos González Pérez — Dirección de Tecnología

Fecha: 22/10/2025

Versión: 1.0

# 1. Introducción

El módulo de Seguridad de ASSIP‑ERP garantiza la autenticación, autorización y auditoría centralizada en todo el ecosistema del ERP. Su objetivo es proteger los datos y operaciones de los usuarios bajo un esquema robusto basado en JWT (JSON Web Token), roles, permisos y registros de auditoría.

# 2. Arquitectura General

El sistema está desarrollado en Spring Boot, con arquitectura modular separada por capas (domain, service, repository, web, config). La autenticación se maneja mediante JWT y la autorización mediante roles y permisos almacenados en la base de datos PostgreSQL en el esquema 'seguridad'.

# 3. Autenticación con JWT

El proceso de autenticación se realiza a través del endpoint /auth/login. El usuario envía sus credenciales (username y password) y, si son válidas, el servidor genera un token JWT firmado con una clave secreta.

Cada solicitud subsecuente a los endpoints protegidos debe incluir el token en el encabezado Authorization: Bearer <token>.

El token tiene una duración configurable y contiene la identidad del usuario, lo que permite mantener sesiones stateless.

# 4. Auditoría y Log de Eventos

El módulo registra automáticamente los eventos críticos (como inicios de sesión o creación de usuarios) en la tabla seguridad.log\_evento. Cada registro incluye los siguientes campos:  
• id\_usuario  
• módulo  
• acción  
• descripción  
• ip\_origen  
• user\_agent  
• fecha\_evento

Ejemplo real:  
1 | LOGIN | Inicio de sesión exitoso | 2025‑10‑22 08:05:39  
4 | CREAR\_USUARIO | Se creó el usuario prueba1 | 2025‑10‑22 08:15:35

# 5. Roles, Permisos y Usuarios

El sistema define tres entidades principales:  
  
• Rol: agrupa permisos y define responsabilidades (ADMIN, SUPERVISOR, LECTURA, OPERATIVO).  
• Permiso: representa acciones específicas como HOJAVIDA\_VIEW, CONTABILIDAD\_EDIT, USUARIOS\_EDIT.  
• Usuario: vinculado a un rol, posee credenciales cifradas y estado activo/inactivo.

# 6. Servicios Principales

• AuthService: gestiona el inicio de sesión y generación de tokens.  
• JwtService: genera y valida los JWT.  
• UsuarioService: CRUD de usuarios y encriptación de contraseñas con BCrypt.  
• RolService: CRUD de roles.  
• PermisoService: CRUD de permisos y asociación con roles.  
• LogEventoService: registra automáticamente acciones relevantes.

# 7. Flujo de Autenticación y Registro

1️⃣ El cliente realiza una petición POST a /auth/login con las credenciales.  
2️⃣ AuthService valida las credenciales en la base de datos (usuarios.username y password encriptado).  
3️⃣ JwtService genera un token JWT.  
4️⃣ JwtAuthenticationFilter valida el token en cada request.  
5️⃣ LogEventoService guarda el registro en seguridad.log\_evento con IP y agente.

# 8. Endpoints Principales

/auth/login – Autenticación y generación de token JWT  
/auth/register – Registro de usuarios  
/seguridad/usuarios – CRUD de usuarios  
/seguridad/roles – CRUD de roles  
/seguridad/permisos – CRUD de permisos  
/seguridad/roles/ping – Verificación de token

# 9. Estado Actual

✅ Login con token JWT funcional  
✅ Auditoría de eventos activa (log\_evento)  
✅ CRUD de usuarios, roles y permisos operativo  
✅ Tokens validados por JwtAuthenticationFilter  
  
Próximos pasos:  
• Tabla relacional rol\_permisos  
• Control de acceso granular por permiso  
• Implementar 2FA para autenticación avanzada.