



MANUAL TÉCNICO PARA LA APLICACIÓN SOMTEQ

El objetivo del presente manual técnico es proporcionar una guía detallada sobre la instalación, configuración, mantenimiento y resolución de problemas del sistema SOMTEQ. Este documento está dirigido al equipo técnico encargado de administrar y dar soporte a la solución.

Introducción

SOMTEQ está diseñado como una plataforma de gestión modular basada en principios de escalabilidad, seguridad y facilidad de uso.

Arquitectura del Sistema

Descripción General

SOMTEQ está basado en una arquitectura cliente-servidor. La interfaz de usuario (frontend) está alojada en un navegador web y se conecta al backend a través de APIs RESTful para la gestión de datos.

Componentes Principales

- Módulo de Inventario: Permite registrar y gestionar equipos y piezas.
- Módulo de Informes: Genera reportes personalizados basados en criterios específicos.
- Módulo de Gestión de Usuarios: Administra roles, permisos y autenticación.
- API RESTful: Gestiona la comunicación entre el frontend y el backend.
- Sistema de Seguridad: Maneja la autenticación y autorización basada en roles.

Instalación y Configuración

Requisitos del Servidor

- Sistema Operativo: Linux (Ubuntu 20.04 o superior) o Windows 10.
- Hardware:
 - CPU: Procesador de 4 núcleos o superior.
 - RAM: Mínimo 8 GB.
 - Almacenamiento: 100 GB libres.
- Software Necesario:
 - Node.js v16+ o Python 3.9+.
 - MongoDB o MySQL 8.0+.
 - Nginx o Apache para el servidor web.

Instalación del Sistema

1. Preparar el entorno:
 - Instalar Node.js según el backend seleccionado.
 - Configurar MySQL con los usuarios y permisos necesarios.
2. Descargar el Código Fuente:
 - Clonar el repositorio desde el control de versiones:
`git clone https://github.com/empresa/somteq.git`
 - `cd somteq`
3. Instalar Dependencias:
 - Backend:
 - `cd backend`
 - `npm install` # Para Node.js
 - `pip install -r requirements.txt` # Para Python
 - Frontend:
 - `cd frontend`
 - `npm install`
4. Configurar Variables de Entorno: Crear un archivo `.env` con las credenciales y parámetros clave.
Ejemplo:
5. `DATABASE_URL=postgresql://usuario:contraseña@localhost:5432/somteq`
6. `JWT_SECRET=clave_secreta`
7. `STORAGE_PATH=/ruta/para/archivos`
8. Iniciar Servicios:
 - Backend:
 - `npm run start` # Node.js
 - `python manage.py runserver` # Django
 - Frontend:
 - `npm run build`

Configuración Inicial

Cargar datos básicos: Usa un script para insertar datos iniciales en la base de datos.

Crear usuario administrador:

```
npm run create-admin
```

Estructura de Base de Datos

Tablas Principales

Usuarios	id_usuario, nombre, correo, contraseña, rol_id
Inventario	id_equipo, nombre, codigo, estado, ubicacion, cantidad
Historial de Cambios	id_cambio, fecha, usuario_id, descripcion

Relación entre Tablas

- Relación 1:N entre Usuarios y Historial de Cambios.
- Relación 1:N entre Inventario y Historial de Cambios.

Gestión de Roles y Permisos

Estructura de Permisos

Cada rol tiene un conjunto de permisos, almacenados en una tabla Roles:

- Administrador: Acceso total.
- Usuario Avanzado: Acceso parcial (sin eliminación).
- Usuario Básico: Solo consulta.

Mantenimiento del Sistema

Respaldo de la Base de Datos

1. Usar herramientas nativas como `pg_dump` para PostgreSQL:
2. `pg_dump -U usuario -d somteq > backup.sql`

Actualizaciones del Sistema

1. Descargar la nueva versión del repositorio:
2. `git pull origin main`
3. Ejecutar migraciones de la base de datos:
4. `npm run migrate`

Seguridad y Buenas Prácticas

- Autenticación: Usar JWT para sesiones seguras.
- Conexión Segura: Configurar HTTPS con certificados SSL.
- Respaldo: Realizar copias de seguridad cada 24 horas.
- Monitoreo: Implementar herramientas como Prometheus o Datadog.

Resolución de Problemas

El sistema no carga correctamente.

1. Verifica que los servicios (backend y base de datos) estén en ejecución.
2. Revisa los logs:
3. `tail -f logs/backend.log`

Errores en la Base de Datos.

1. Asegúrate de que las migraciones estén al día:
2. `npm run migrate`