Mapa de proyecto

Sistema de Seguimiento de Transporte en Tiempo Real

Documento de Requerimientos – Sistema de Seguimiento de Transporte en Tiempo Real

1. Objetivo General

Construir un sistema de seguimiento en tiempo real de vehículos de transporte, que permita a diferentes entidades de transporte (públicas o privadas) registrar, gestionar y visualizar sus flotas, y que a su vez los usuarios finales puedan consultar la ubicación de los vehículos mediante una aplicación móvil.

---

2. Roles Principales

1. **Administrador (web – React):**
    - Gestión de entidades de transporte.
    - Registro y administración de vehículos.
    - Asociación de conductores con vehículos.
    - Definición de visibilidad de la entidad (pública o privada).
    - Visualización en mapa de las posiciones de vehículos en tiempo real.
2. **Conductor (móvil – Flutter, app simple):**
    - Inicia sesión con sus credenciales o código.
    - Activa su vehículo (en servicio).
    - Envia periódicamente su ubicación (latitud/longitud) al backend.
3. **Usuario Público (móvil – Flutter, app cliente):**
    - Consulta entidades disponibles (públicas o privadas).
    - Accede a la entidad seleccionada:
        - Si es pública → acceso directo.
        - Si es privada → acceso mediante código de invitación.
    - Visualiza en mapa los vehículos activos en tiempo real.

---

3. Funcionalidades

3.1 Backend (Spring Boot + PostgreSQL)

- Gestión de usuarios y roles: administrador, conductor, usuario público.
- Gestión de entidades de transporte: crear, modificar, definir tipo (pública/privada).
- Gestión de vehículos: registrar vehículos, asociarlos a una entidad y un conductor.
- Ubicaciones en tiempo real: endpoint para recibir lat/long del conductor.
- Visualización de ubicaciones: endpoints para servir datos al admin y al usuario público.
- **Seguridad básica:**
  - Primera versión → tokens simples/códigos de acceso.
  - Futuro → integración con Keycloak u OAuth2.
- **Historial de posiciones:** almacenar recorrido con timestamp (mínimo opcional en MVP).

3.2 Administración (React)

- Login de administrador.
- Panel de control con:
  - Gestión de entidades de transporte.
  - Gestión de vehículos y conductores.
  - Vista en mapa (Leaflet.js u otra) de los vehículos activos.
- Opciones de entidad pública/privada con generación de código de acceso.

3.3 Aplicación de Conductor (Flutter – simple)

- Login o acceso con código de asignación.
- Activar vehículo en servicio.
- Enviar ubicación periódicamente al backend.
- Indicador de estado (en servicio / fuera de servicio).

3.4 Aplicación Cliente Público (Flutter – móvil)

- Lista de entidades de transporte disponibles.
- Acceso a entidades públicas directamente.
- Acceso a entidades privadas con código.
- Mapa en tiempo real mostrando vehículos activos.

---

4. Requisitos No Funcionales

- Escalabilidad: soporte para múltiples entidades y flotas.
- Disponibilidad: backend y DB en contenedores (Docker).
- Usabilidad: interfaz clara tanto en web (admin) como en apps móviles.
- Compatibilidad: PostgreSQL como motor principal.
- **Seguridad:**
  - Nivel básico inicial: acceso con credenciales simples y códigos de invitación.
  - Nivel avanzado futuro: Keycloak + OAuth2.
- **Mantenibilidad:** uso de logs, perfiles (dev, prod) y documentación (Swagger).

---

5. Tecnologías Base

- Backend: Spring Boot, Maven, PostgreSQL, JPA, Liquibase, Swagger.
- Frontend Admin: React + Leaflet.js (mapas).
- App Conductor: Flutter (simple, solo envío de ubicación).
- App Usuario: Flutter (visualización de ubicaciones en mapa).
- Infraestructura: Docker (DB + backend), Postman (pruebas).

---

6. Flujo de Datos Simplificado

1. Conductor → App Flutter → envía lat/long → Backend.
2. Backend → guarda en PostgreSQL + expone endpoint.

3. Administrador / Usuario público → App React/Flutter → consulta API → muestra mapa.

---

7. Roadmap de Implementación

1. PostgreSQL: diseño del modelo inicial (usuarios, entidades, vehículos, ubicaciones).
2. Spring Boot: endpoints CRUD + conexión DB.
3. React (Admin): panel básico + mapa.
4. Flutter (Conductor): app mínima de envío de ubicación.
5. Flutter (Usuario): app básica de visualización.
6. Contenedores + Postman: pruebas y despliegue.
7. Extras: seguridad avanzada, historial de rutas, notificaciones.