

A continuación se presenta un ejemplo de plan de desarrollo y lista de funcionalidades para la aplicación, considerando el Manifiesto Anthem y aprovechando el extenso dataset proporcionado.

Lista de Funcionalidades

1. Autenticación y Gestión de Usuarios

- Registro e inicio de sesión (roles: administrador, analista, ciudadano).
- Panel de administración para gestionar permisos y configuraciones.

2. Dashboards y Visualización de Datos

- Paneles interactivos con gráficos y estadísticas para cada área (calidad del aire, contaminación acústica, tráfico, accidentes, etc.).
- Filtros y selección de periodos para análisis detallado.

3. Mapa Interactivo y Geolocalización

- Visualización de sensores y puntos de interés (estaciones de tráfico, contenedores, paradas de taxi, ubicaciones de patinetes/bicicletas, etc.) sobre mapas.
- Integración con servicios de geolocalización (por ejemplo, Leaflet o Google Maps).

4. Análisis Predictivo y Alertas

- Implementación de modelos básicos (reglas o machine learning) para identificar patrones en tráfico, accidentes y consumo energético.
- Generación de alertas y recomendaciones en tiempo real para la toma de decisiones.

5. Optimización de Movilidad y Gestión Urbana

- Funcionalidades para sugerir rutas alternativas y mejoras en la movilidad inteligente.
- Gestión de incidencias: reporte y seguimiento de accidentes, infracciones y denuncias.

6. Gestión Ecológica y Sostenibilidad

- Monitorización en tiempo real de la calidad del aire y niveles de contaminación acústica.
- Gestión y optimización de la generación y uso de energías renovables (por ejemplo, datos de instalaciones fotovoltaicas).

7. Integración y Procesamiento de Datos

- API REST para consumir y procesar datos de los diferentes sensores y fuentes (trafico, censo, multas, etc.).
- Conexión con MongoDB para almacenar y consultar los datos de forma eficiente.

Plan de Desarrollo para un Equipo de Dos Personas

Fase 1: Análisis y Diseño (1 semana)

- **Reunión inicial:** Repasar el Manifiesto Anthem y el dataset para identificar las áreas clave.
- **Diseño de la arquitectura:**
 - Definir la estructura de la aplicación (separación de backend y frontend).
 - Diseñar el modelo de datos en MongoDB y la estructura de las API REST.
 - Bocetar wireframes y flujos de usuario para los dashboards y mapas.

Fase 2: Configuración del Entorno y Herramientas (1 semana)

- Configurar el repositorio de código (Git) y establecer un flujo de trabajo (branching, code review).
- Configurar entornos de desarrollo, pruebas y despliegue (puede incluir CI/CD básico).
- Elegir frameworks y librerías:
 - Backend: Node.js con Express.
 - Frontend: React, integrando librerías para gráficos (Chart.js, D3.js) y mapas (Leaflet, Google Maps API).
 - Base de datos: MongoDB.

Fase 3: Desarrollo del Backend (2-3 semanas)

- **Persona A (Backend):**
 - Implementar la estructura del servidor con Node.js y Express.
 - Definir endpoints para la gestión de usuarios, autenticación y consumo de datos de los sensores.
 - Modelar y conectar la base de datos MongoDB (colecciones para cada tipo de sensor/dato).
 - Desarrollar servicios para procesar y normalizar los datos del dataset.

Fase 4: Desarrollo del Frontend (2-3 semanas, en paralelo con el Backend)

- **Persona B (Frontend):**
 - Configurar el proyecto React y la comunicación con la API REST.
 - Desarrollar la interfaz de usuario:
 - Página de autenticación y panel de usuario.
 - Dashboards interactivos que muestren gráficos y estadísticas de las distintas áreas.
 - Mapa interactivo para visualizar los datos geolocalizados.
 - Integrar componentes de análisis predictivo y alertas en la interfaz.

Fase 5: Integración y Pruebas (1-2 semanas)

- Integrar las partes desarrolladas (backend y frontend).
- Realizar pruebas unitarias y de integración en ambos lados.
- Validar la precisión de los datos mostrados y la respuesta de la aplicación ante diferentes escenarios (p.ej., alertas en tiempo real, consulta de datos históricos).

Fase 6: Despliegue y Monitorización (1 semana)

- Configurar el despliegue en un servidor o plataforma en la nube (Heroku, AWS, etc.).
- Implementar herramientas de monitorización para el rendimiento y la seguridad de la aplicación.
- Planificar una fase beta con usuarios reales para recoger feedback y ajustar funcionalidades.

Conclusión

Este plan y conjunto de funcionalidades no solo abordan el cumplimiento del Manifiesto Anthem (optimizando movilidad, seguridad, sostenibilidad y calidad de vida), sino que también aprovechan el rico dataset para generar insights valiosos. La división en backend y frontend permite que dos desarrolladores colaboren de forma paralela, asegurando un desarrollo ágil y coordinado.

Esta estrategia proporciona una base sólida para construir una solución integral que pueda adaptarse y escalar conforme se vayan identificando nuevas necesidades o se amplíen los datos disponibles