

# URBAN PULSE

*Plataforma Inteligente de Movilidad Urbana y Calidad del Aire*

Iñigo Diaz Munio  
79134735S  
CDIA + INF

# *¿Qué es UrbanPulse?*

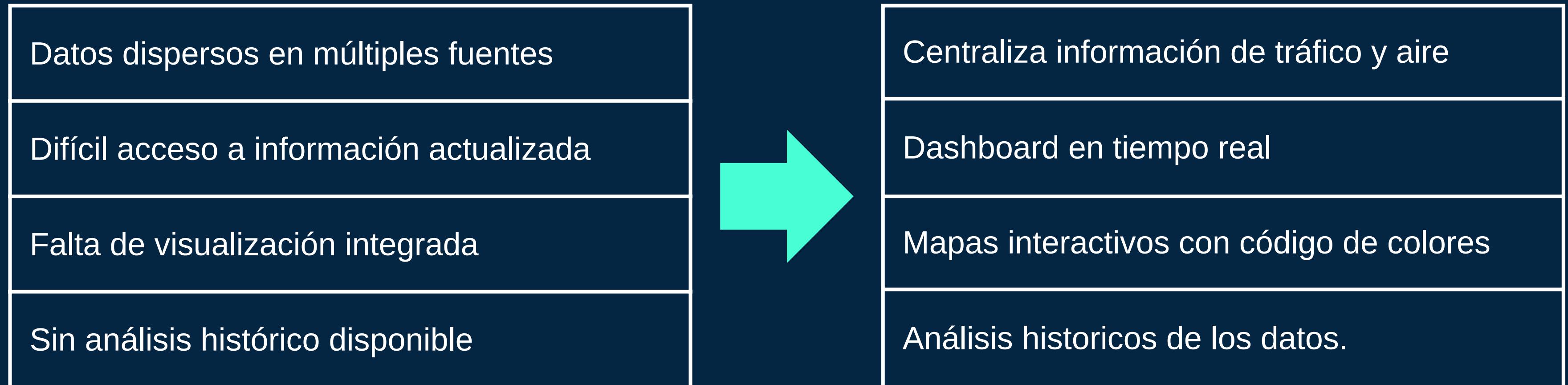
*Plataforma web de análisis de movilidad urbana y calidad del aire en tiempo real.*

*Integra datos oficiales del Gobierno Vasco (Open Data Euskadi).*

*16 municipios principales del Euskadi.*

*Actualización automática cada 5-10 minutos.*

# *Problema y Solución*



# *Arquitectura de Microservicios*

Frontend → *React 18 + TypeScript + Leaflet (mapas).*

API Gateway → *Node.js + Express + JWT*

*Dos microservicios*

Microservicio Tráfico	Microservicio Calidad del Aire
Node.js + MongoDB	FastAPI + PostgreSQL



*Todo orquestado con Docker Compose*

# *Funcionalidades Principales*

**1-Dashboard en Tiempo Real con mapa interactivo de Euskadi.**

*Métricas en tiempo real: Vehículos/hora, PM10, NO2, O3, incidencias*

*Código de colores por nivel de contaminación*

*Filtros dinámicos de contaminantes*

*Gráficos comparativos por provincias (Bizkaia, Gipuzkoa, Araba)*

**2-API REST documentada con Swagger/OpenAPI 3.0**

**3-Autenticación JWT con roles (user/admin)**

*Usuarios @urbanpulse.com = Administradores automáticos*

# Tecnologías

## ARQUITECTURA

*Microservicios desacoplados*  
*API REST + Swagger/OpenAPI*

3.0

*Autenticación JWT con roles*

## DEV Ops

*Docker Compose*

*Nginx*

*Healthchecks Automaticos*

## BACKEND

*Node.js 18 + Express*  
*FastAPI (Python 3.10)*  
*JWT*

## FRONTEND

*React 18.2.0*  
*TypeScript 5.6.2*  
*Vite 5.0.0*  
*Recharts (gráficos)*  
*React-Leaflet (mapas)*

## BASE DE DATOS

*MongoDB 6.0*  
*PostgreSQL 15*

# *Fuente de datos + Deployment y DevOps*

*Open Data Euskadi - Datos de tráfico en tiempo real*

*Red de Control de Calidad del Aire - Gobierno Vasco*

*16 municipios principales de Euskadi.*

*6 contaminantes monitorizados: PM10, PM2.5, NO2, O3, SO2, CO*

# *Fuente de datos + Deployment y DevOps*

```
C:\Users\Iñigo\Desktop\UrbanPulse-DesarrolloAppWeb\UrbanPulse-DesarrolloAppWeb>docker compose up --build -d
```

*docker-compose up --build*

*6 servicios orquestados: MongoDB, PostgreSQL, 2 microservicios, Gateway, Frontend*

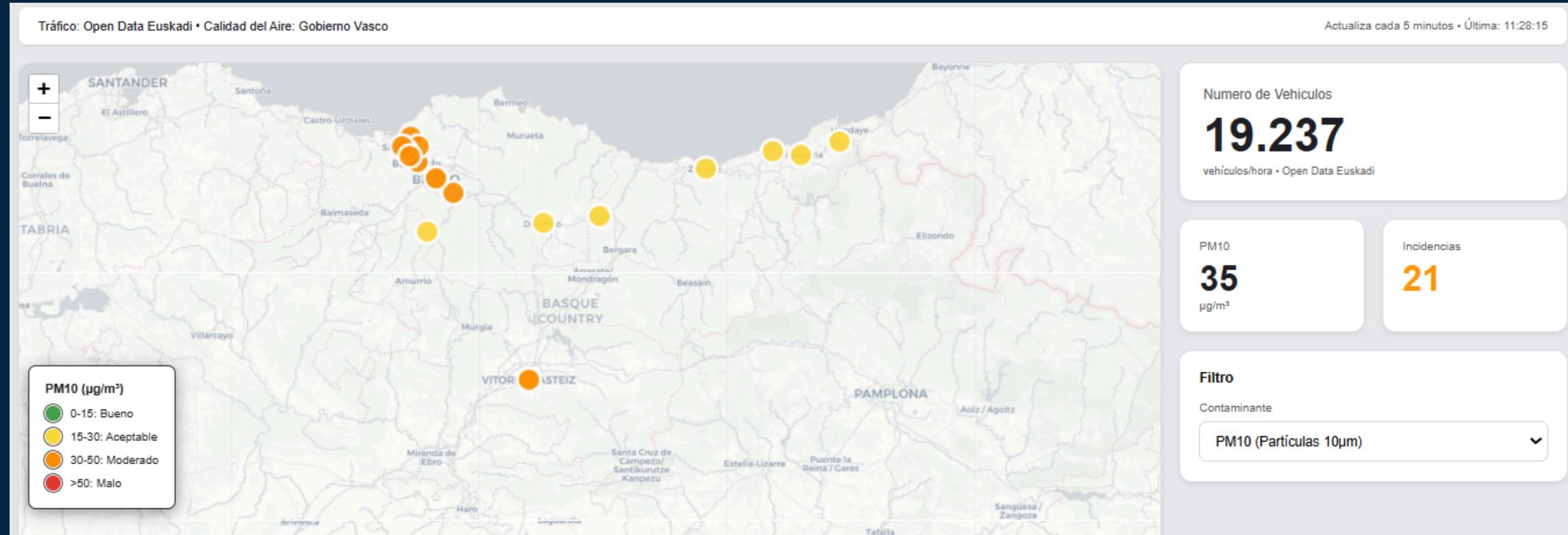
*Healthchecks automáticos para todos los servicios*

*Volúmenes persistentes para bases de datos*

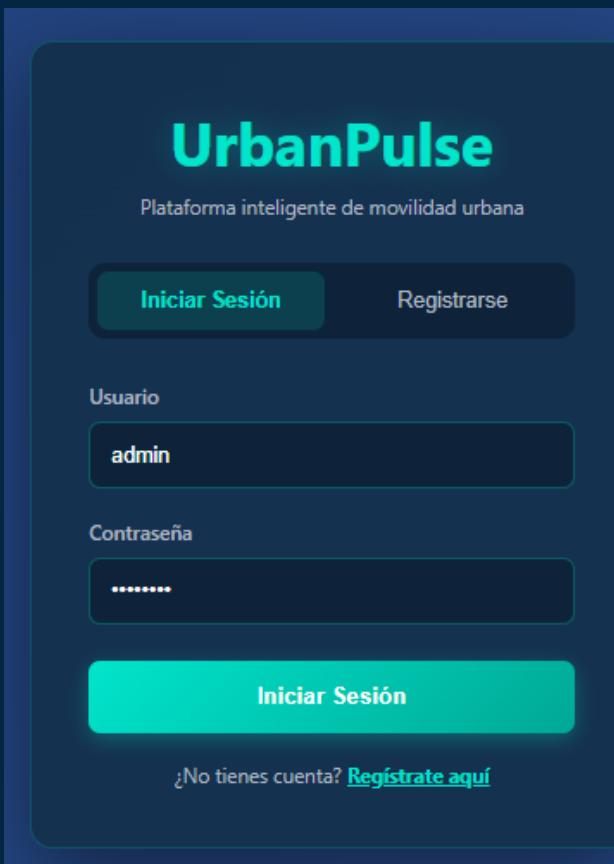
*Nginx para frontend estático*

# Resultados

*Dashboard funcional con datos reales de Euskadi*



*Sistema de autenticación*



*Arquitectura escalable*

*Proyecto 100% funcional dockerizado*

## Histórico

Esta página de Histórico permite analizar la evolución del tráfico y la contaminación a lo largo del tiempo.

Zona

Seleccionar zona

Contaminante

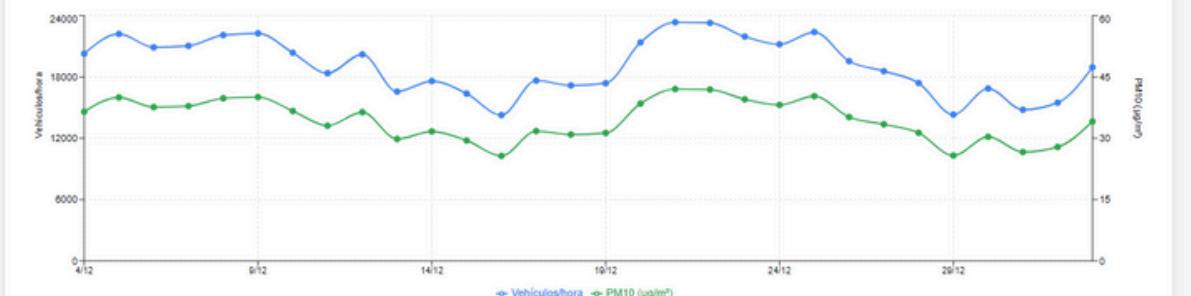
PM10

Fecha (Hasta)

dd/mm/aaaa

Actualizar

### Evolución Temporal - Últimos 30 Días

Mostrando todas las zonas • Contaminante: PM10 ( $\mu\text{g}/\text{m}^3$ )

### Datos Históricos por Zona - Últimos 10 Días.

Fecha	Zona	Tráfico (veh/h)	PM10 ( $\mu\text{g}/\text{m}^3$ )	PM2.5 ( $\mu\text{g}/\text{m}^3$ )	NO2 ( $\mu\text{g}/\text{m}^3$ )	O3 ( $\mu\text{g}/\text{m}^3$ )
24/12/2025	Barakaldo	1727	56	35	72	55

## Panel de Administración

Gestiona usuarios, monitorea servicios y supervisa el sistema UrbanPulse.

### Usuarios (2)

Servicios Logs del Sistema

#### Gestión de Usuarios

Los nuevos usuarios deben registrarse desde la página de login. Usa emails @urbanpulse.com para crear administradores automáticamente.

Usuario	Email	Rol	Fecha Registro	Acciones
admin	admin@urbanpulse.com	Admin	4/12/2025	Cambiar Rol Eliminar
Inigo	inigodm12@gmail.com	Usuario	24/11/2025	Cambiar Rol Eliminar

## REST API

La sección API REST ofrece acceso programático a los datos de UrbanPulse mediante una interfaz documentada con OpenAPI 3.0.

### Endpoints Principales

**GET** /api/movilidad/realtime

Recupera datos de tráfico y movilidad en tiempo real de todas las estaciones de Euskadi.

Respuesta: Array de objetos con zone, vehicle\_count, incident\_count, timestamp, etc.

**GET** /api/meteorologia/realtime

Devuelve niveles de contaminación atmosférica (PM10, PM2.5, NO2, O3, SO2, CO) de todas las estaciones.

Respuesta: Array de mediciones con city, zone, pollutant, value, unit, timestamp.

**POST** /api/login

Permite autenticar usuarios y obtener un token JWT para acceder a los endpoints protegidos.

Body: { "username": "usuario", "password": "contraseña" }

Respuesta: { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..." }

**POST** /api/registro

Permite registrar nuevos usuarios en el sistema.

Body: { "username": "nuevo\_usuario", "password": "contraseña\_segura" }

# Conclusiones

## **LOGROS**

*Arquitectura de microservicios.*  
*Integración con APIs oficiales.*  
*Sistema de autenticación (JWT).*  
*Persistencia dual (NoSQL + SQL).*  
*Deployment con Docker.*

## **COBERTURA**

*16 municipios de Euskadi.*  
*Datos del Gobierno Vasco.*  
*6 contaminantes monitorizados.*  
*Actualización automática.*  
*Interfaz intuitiva y accesible.*

## **APRENDIZAJES CLAVE**

*Dominio de arquitectura full-stack  
(React + Node.js + FastAPI).*  
*Uso de bases de datos relacionales y  
no relacionales.*  
*Implementación de CI/CD y  
contenedорización con Docker.*

## **ESCALABILIDAD**

*Proyecto listo para expansión:*  
*Más municipios, posibles  
predicciones con ML, más APIs  
públicas...*