

# Memoria del Proyecto: Aplicación de Reportes Urbanos

Durante el desarrollo de este proyecto he creado una aplicación web que permite a los ciudadanos reportar incidencias urbanas de forma sencilla y rápida. Esta aplicación está pensada para facilitar la comunicación entre los usuarios y los servicios de mantenimiento del entorno urbano.

## Objetivo

El objetivo principal era construir una plataforma funcional en la que un usuario pudiera:

- Registrarse e iniciar sesión.
- Enviar reportes con título, descripción, ubicación y estado.
- Adjuntar imágenes para ilustrar cada incidencia.
- Consultar los reportes existentes.
- Editar o eliminar sus propios reportes si está autenticado.
- Exportar todos los reportes a formato CSV.
- Navegar cómodamente gracias a la paginación.

## Tecnologías utilizadas

- **Frontend:** React con Tailwind CSS.
- **Backend:** Django y Django REST Framework.
- **Base de datos:** SQLite.
- **Control de versiones:** Git.

## Estructura del proyecto

Dividí el proyecto en dos partes principales:

1. **Backend (Backend\_reportes):** Aquí implementé la API con Django REST Framework. Definí el modelo Reporte con los campos necesarios (título, descripción, ubicación, estado, fecha e imagen). También configuré la autenticación de usuarios para permitir login y registro desde el frontend.
2. **Frontend (Fronted):** Utilicé React para construir la interfaz de usuario. Me apoyé en axios para conectar con la API y usé react - icons para añadir iconos visuales. Tailwind CSS me permitió maquetar la aplicación de forma moderna y responsive.

## Gestión de la base de datos

No fue necesario crear manualmente la base de datos ni generar scripts .sql, ya que Django se encarga de todo esto automáticamente con el comando migrate.

La base de datos cuenta con una tabla principal (Reporte) que guarda todos los datos enviados por los usuarios, y además se apoya en el sistema de usuarios propio de Django.

## Dificultades encontradas

Durante el desarrollo me encontré con algunos problemas técnicos:

- **Incompatibilidades entre versiones de React y algunas dependencias.** Lo solucioné utilizando el parámetro `--legacy-peer-deps` al instalar los paquetes.
- **Problemas visuales con el margen de los botones de paginación.** Me llevó bastante tiempo ajustar correctamente el espaciado entre botones y el diseño final, probando tanto en el JSX como en los estilos de CSS.
- **Configuración de subida de imágenes:** Tuve que trabajar con `FormData` en el frontend y configurar correctamente los headers para que Django aceptara las imágenes.

Estoy satisfecho con el resultado final. La aplicación funciona correctamente y cubre todas las funcionalidades que me propuse desde el principio. Además, ha sido una experiencia muy útil para afianzar mis conocimientos sobre desarrollo web full stack y para resolver errores reales que me obligaron a aprender nuevas soluciones y herramientas



