

### INTRODUCCIÓN

Hasta hace poco muchos vehículos circulaban por las grandes ciudades superando los límites de contaminación establecidos. Además, sumado a los gases que emiten las grandes empresas en las ciudades, ha provocado que se hayan establecido una serie de medidas o restricciones: como el control del tráfico, el uso de coches con menos emisiones de (CO<sub>2</sub>, NO<sub>X</sub>, CO...), habilitar más zonas peatonales, fomentar el transporte público ecológico, usar más la bicicleta por zonas destinadas a esta práctica...etc

### OBJETIVO

El objetivo principal del proyecto es desarrollar una aplicación que permita conocer la información de concentración de polución y contaminación en grandes núcleos urbanos de forma diaria. El acceso a esta información permitirá notificar a los usuarios las restricciones de circulación existentes en el casco urbano de estas localidades.

Para la obtención del objetivo principal será necesario satisfacer los siguientes objetivos secundarios:

- Registrar usuarios y cada usuario podrá tener uno o más vehículos registrados.
- A través de un mapa de Europa mostrar en que zonas puede circular dependiendo del tipo de vehículo registrado. Mostrar la ubicación en tiempo real del usuario.
- Acceder a información más detallada sobre una zona restringida (Datos de las partículas del aire actuales, restricciones y prohibiciones establecidas, tipo de zona de bajas emisiones, histórico de las partículas del aire, precio de la multa por incumplimiento...)

### DESCRIPCIÓN TÉCNICA

Se usará una arquitectura **cliente-servidor (MVC)**. Para el back-end utilizaré NodeJS, MongoDB, ExpressJS, y algún framework para el front-end como Bootstrap, Angular, JQuery...

**Nodejs:** Se usará como lenguaje de programación JavaScript, en este caso NodeJS, que nos permitirá trabajar en el back-end tanto del lado del cliente como del servidor sin tener que usar otros medios. Además, trabaja a buen rendimiento, es altamente escalable, maneja alto tráfico de usuarios o eventos y disminuye el margen de aparición de errores.

**MongoDB:** Como base de datos no relacional que almacena objetos en formato JSON.

**ExpressJS:** Es un framework sobre NodeJS que nos permite trabajar con el protocolo http y tener sistemas de rutas.

**Diseño:** Para el diseño se usará algunos de los frameworks para el front-end como puede ser Angular para la parte del cliente, Bootstrap...

**Librería mapa:** Se usará alguna librería que implemente un mapa base para la geolocalización o posicionamiento de los datos como: Leaflet, Api Google Maps...

**Documentación:** Se usará el lenguaje de modelado de software (UML)

### SOFTWARE

**Documentación:** Word(Texto), Excel (Diagrama de Gantt), Visual Paradigm (Modelado) y Trello (Planificación).

**Programación:** Sublime Text 3 (Código), Robo 3T (Gestor base de datos), Postman (Testear) y GitHub (Almacenamiento y mantenimiento).