

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**Grado en Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**APLICACIÓN PARA LA SIMULACIÓN Y PREDICCIÓN  
DE RUTAS MENOS CONTAMINANTES**

**Miguel García García  
Tutor: Inmaculada Luengo  
Ponente: Iván González Martínez**

**Julio 2018**



# **APLICACIÓN PARA LA SIMULACIÓN Y PREDICCIÓN DE RUTAS MENOS CONTAMINANTES**

**AUTOR: Miguel García García**  
**TUTOR: Inmaculada Luengo**

**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Julio de 2018**



# Resumen (castellano)

Este Trabajo Fin de Grado (en adelante TFG) sirve como trabajo para los desarrollos que tendrán lugar en el proyecto europeo ESTABLISH (Environmental Sensing To Act for a Better quality of Life: Smart Health).

El objetivo de este proyecto es el desarrollo de un servicio de predicción que permita determinar la ruta menos contaminante entre dos puntos, teniendo en cuenta factores como el tipo de jornada (si es laborable o no) y la climatología. Dichos factores son obtenidos de datos reales, los cuales se han ido almacenados durante un largo periodo de tiempo gracias a diferentes open data, de manera que se pretende alcanzar la máxima veracidad posible.

Para alcanzar la predicción el servicio se apoya en el uso de simulaciones personalizadas con los factores comentados previamente.

Aunque el desarrollo de este proyecto podría centrarse en cualquier ciudad en particular, en este caso las ciudades con las que se ha trabajado han sido las españolas Madrid y Valencia, siendo esta última la que más avanzada en el desarrollo ha quedado.

Para llevar a cabo el TFG he empleado muchos conocimientos adquiridos en mi etapa de aprendizaje del grado, si bien he tenido que formarme en el uso de otras muchas. Esto queda detallado para cada una de las tecnologías a lo largo del documento.

Finalmente se pretende que los ayuntamientos de las ciudades puedan utilizar esta herramienta en vistas de conseguir con antelación modelos preventivos contra la contaminación, aplicable a la localidad, logrando evitar adoptar medidas más extremas cómo las actuales.

En el siguiente documento se muestran todas las etapas por las que ha pasado el proyecto, desde el inicio hasta llegar al momento actual, ya que se trata de un proyecto que a día de hoy está en desarrollo.

# Palabras clave (castellano)

Open data, simulación, veracidad, tecnologías.

# **Abstract (English)**

This Bachelor Thesis (hereinafter TFG) serves as work for the developments that will take place in the European project ESTABLISH (Environmental Sensing To Act for a Better Quality of Life: Smart Health).

The objective of this project is the development of a prediction service that allows determining the least polluting route between two points, taking into account factors such as the type of working day (whether it is working or not) and the weather. These factors are obtained from real data, which have been stored for a long period of time thanks to different open data, so that it is intended to reach the maximum possible veracity.

To reach the prediction, the service relies on the use of personalized simulations with the previously mentioned factors.

Although the development of this project could focus on any particular city, in this case the cities with which it has worked have been the Spanish Madrid and Valencia, the latter being the most advanced in the development has been.

To carry out the TFG I have used many acquired knowledge in my stage of learning the degree, although I have had to train in the use of many others. This is detailed for each of the technologies throughout the document.

Finally, it is intended that city councils can use this tool in advance to get preventive models against pollution, applicable to the locality, being able to avoid adopting more extreme measures like the current ones.

The following document shows all the stages through which the project has passed, from the beginning to the current moment, since it is a project that is currently under development.

# **Keywords (inglés)**

Open data, simulation, veracity, technologies.

## *Agradecimientos*

A la Universidad Autónoma de Madrid por haberme dado la formación necesaria para convertirme en un Ingeniero informático.

A mis compañeros de la universidad, con los cuales he compartido buenos y malos momentos en todos estos años y de los cuales, algunos me los llevo como amigos.

A mis amigos, quienes han constituido un soporte importantísimo para mí en este tiempo. Gracias a ellos he podido superar malas rachas, de las cuales ha habido unas cuantas, y convertirme en lo que soy hoy.

A Hiberia, por haberme dado la oportunidad de llevar a cabo este proyecto, en especial a Inmaculada Luengo por ser mi tutora, aconsejándome siempre que lo he requerido en este proyecto y a mi jefe de proyecto Samuel Méndez, del cual he aprendido muchísimo y que ha supuesto una grandísima ayuda para mi adaptación, ayudándome con cualquier problema que me surgiese.

A mi ponente Iván González Martínez. No me puso ninguna pega cuando acudí a el solicitándoselo. Siempre ha estado disponible cuando lo he necesitado.

Y por último y más importante, a mi familia. Puedo afirmar al cien por cien que sin ellos no lo habría conseguido. Son lo más importante que tengo y siempre han estado ahí para apoyarme, en especial en los momentos más complicados en los que estuve a punto de desistir, instante donde más empujaron.

En definitiva, a todos los que han aportado su granito de arena para hacer que consiga lo que hace unos años nadie hubiese pensado. Por todo ello... ¡MIL GRACIAS!





## INDICE DE CONTENIDOS

1	Introducción .....	1
1.1	Motivación .....	1
1.2	Objetivos .....	2
1.3	Organización de la memoria .....	2
2	Estado del arte .....	3
2.1	Introducción .....	3
2.2	Situación actual .....	3
2.2.1	Life Gystra .....	3
2.2.2	Clean Space .....	4
2.2.3	ALD Ecodrive .....	4
3	Diseño .....	5
3.1	Diseño de la aplicación .....	5
3.2	Datasets .....	5
3.3	Diagrama .....	7
4	Desarrollo .....	8
4.1	Introducción .....	8
4.2	Mapas .....	8
4.2.1	Valencia .....	9
4.2.1.1	Obtención datasets .....	9
4.2.2	Madrid .....	10
4.2.2.1	Obtención datasets .....	10
4.3	Infraestructura de desarrollo .....	14
4.3.1	Hardware .....	14
4.3.2	Software .....	14
4.4	Lenguajes de programación .....	15
4.4.1	Python .....	15
4.4.2	Scripting .....	15
4.4.3	Query DSL .....	15
4.5	Herramientas de desarrollo .....	16
4.5.1	Elasticsearch .....	16
4.5.2	MongoDB .....	17
4.5.3	Robo3T .....	17
4.5.4	API Rest .....	17
4.5.5	Cron .....	17
4.5.5.1	Crontab .....	17
4.5.6	Rdesktop .....	18
4.5.7	SUMO (Simulation of Urban MOvility) .....	18
4.5.7.1	RandomTrips .....	18
4.5.7.2	Duarouter .....	18
4.5.7.3	Sumo-Gui .....	19
4.5.8	Google Maps .....	19
4.6	Estudio del flujo de la circulación .....	20
4.7	Colocación de las estaciones de tráfico .....	22
4.8	Colocación de las estaciones de contaminación .....	23
4.9	Obtención de los diferentes modelos de día .....	23
5	Integración, pruebas y resultados .....	25
6	Conclusiones y trabajo futuro .....	31

6.1 Conclusiones.....	31
6.2 Trabajo futuro .....	31
Referencias.....	33
Glosario.....	35

## INDICE DE FIGURAS

Figura 1: Niño sufriendo contaminación. ....	1
Figura 2: Logo HI Iberia.....	3
Figura 3: Life GySTRA .....	3
Figura 4: Clean Space .....	4
Figura 5: ALD Automotive.....	4
Figuras 6, 7: Datasets.....	6
Figura 8: Diagrama de diseño.....	7
Figura 9: Mapa de Valencia.....	9
Figura 10: Mapa de Madrid .....	10
Figura 11: Fichero Crontab.....	14
Figura 12: Logo SUMO.....	18
Figura 13: Google Maps. ....	19
Figura 14: Distritos de Valencia. ....	20
Figura 15: ID de las calles. ....	21
Figura 16: Ruta óptima .....	21
Figura 17: Estaciones de tráfico.....	22
Figura 18: Modelo de día.....	24
Figura 19: Salida del programa.....	25
Figura 20: Intensidad de tráfico en Valencia.....	26

## INDICE DE TABLAS

Tabla 1: Open Data Madrid.....	13
Tabla 2: Datos objetivo.....	26
Tabla 3: Simulación por defecto.....	27
Tabla 4: Simulación 1.....	27
Tabla 5: Simulación 2.....	28
Tabla 6: Simulación 3.....	28
Tabla 7: Simulación 4.....	28
Tabla 8: Simulación 5.....	29
Tabla 9: Simulación 6.....	29

# 1 Introducción

---

## 1.1 Motivación

487000, esa es la cifra que la Agencia Europea de Medio Ambiente (AEMA) ha proporcionado como la cantidad de muertes causadas al año por contaminación atmosférica en el viejo continente. Si hablamos de España el número es de 31520 fallecidos. [\[1\]](#)



Figura 1: Niño sufriendo contaminación.

Precisamente dos de los principales contaminantes del aire son originados por el uso del vehículo. El primero son las partículas en suspensión (principalmente de los diésel), las cuales se consideran el problema más grave en cuanto a contaminación del aire se refiere, ya que se relacionan con enfermedades respiratorias cardiovasculares y cáncer de pulmón. Previniendo una mínima parte de esta emisión se podrían prevenir unas 1720 muertes al año. Por otra parte tenemos el dióxido de nitrógeno, que interviene en la generación de partículas cuya respiración puede provocar disfunciones en el aparato respiratorio y una menor respuesta del aparato inmunológico.

Si bien es cierto que cada año existe una mayor concienciación respecto a este tema, todavía queda mucho por avanzar. Ciudades como Madrid adoptan protocolos anticontaminantes cuando se alcanzan niveles muy altos como son reducir la velocidad en la m-30 a 70 km/h o prohibir el paso en según qué día a matriculas pares o impares dentro del recinto delimitado por la m-30.

Estas medidas también han provocado el disgusto en parte de la población, puesto que les parecen medidas excesivas.

Todos estos datos se tradujeron en un aliciente para adentrarme en el desarrollo de la aplicación, puesto que le veo potencial para que en un futuro y siempre que se utilice bien, supongan una sustancial reducción de la contaminación causada por la circulación y por consiguiente una posible reducción del número de fallecidos.

## **1.2 Objetivos**

Mi trabajo se ha centrado en el desarrollo de un servicio que permita la predicción de las situaciones de una ciudad cara al tráfico y que posteriormente podrá ser utilizado por una aplicación para el cálculo de la ruta menos contaminante entre diferentes puntos de una ciudad. Para ello se apoya en la realización de simulaciones de tráfico, mediante el uso de información real, que pretenden tener el máximo nivel de veracidad posible, pudiendo predecir los respectivos niveles de contaminación en las calles con distintos modelos de día.

Sin embargo esta aplicación forma parte de algo más grande. El desarrollo de esta herramienta forma parte de un proyecto mayor. La idea es que el día de mañana los ayuntamientos puedan utilizar dicho proyecto para prever situaciones de riesgo y aplicar posibles nuevos protocolos anti contaminantes con antelación, evitando aplicar algunos como los de Madrid que quizá resulten un poco drásticos, suponiendo de esta manera un menor impacto para la sociedad.

## **1.3 Organización de la memoria**

La memoria consta de los siguientes capítulos:

- Capítulo 2- Estado del arte: expongo las principales tecnologías existentes a día de hoy con respecto al tema que trata el proyecto.
- Capítulo 3- Diseño: explicación del diseño llevado a cabo para poder realizar la aplicación.
- Capítulo 4- Desarrollo: argumento todo lo que ha sido necesario para elaborar el proyecto, desde las tecnologías utilizadas hasta el desarrollo de los modulos si cómo los problemas surgidos.
- Capítulo 5- Integración, pruebas y resultados: en esta sección me centro sobre todo en las pruebas y en los resultados del proyecto.
- Capítulo 6- Conclusión y trabajo futuro: explico que ha supuesto para mi llevar a cabo este proyecto, así como los siguientes pasos a dar.

## 2 Estado del arte

---

### 2.1 Introducción

Desde el momento en que en el siglo XIX se crean los primeros vehículos por motor de combustión interna con gasolina se puede decir que surge el problema que este proyecto abarca. Si bien no es hasta bien avanzados en el siglo XX y sobre todo el centenario en el que nos encontramos cuando la sociedad toma conciencia sobre los causados y que se van acrecentando con el tiempo.

Esta aplicación está siendo desarrollada por la empresa HIberia, desde donde han visto una oportunidad para avanzar en este terreno.



Figura 2: Logo HI Iberia

### 2.2 Situación actual

Lo cierto es que a día de hoy nos encontramos en una situación que está empezando a despegar pero en la que, en mi opinión, queda mucho por hacer. Si bien es cierto que existe gran cantidad de aplicaciones que proporcionan datos sobre la contaminación del aire y similares, también hay que destacar la escasez de estas enfocadas a lograr el objetivo que busco con el desarrollo del proyecto. A continuación procedo a comentar algunas de las tecnologías y aplicaciones más importantes.

#### 2.2.1 Life Gystra

Programa pionero en Madrid de medición de la contaminación de los vehículos en tiempo real. Se trata del mayor programa de medición continua de emisiones de vehículos en circulación realizado hasta la fecha en Europa, a partir de tecnologías de teledetección de la contaminación. Tiene como objetivo desarrollar una nueva política de movilidad sostenible basada en información empírica sobre las emisiones procedentes del tráfico de carretera. [\[2\]](#)



Figura 3: Life GySTRA

La tecnología de teledetección permite determinar las concentraciones medias de diferentes gases contaminantes y partículas producidos por cada automóvil en conducción normal. Este sistema (funcionando ya en el Paseo de la Castellana), permite analizar datos de CO, CO<sub>2</sub>, hidrocarburos, NO<sub>x</sub> y partículas, además de datos de velocidad y carga del transporte.

### 2.2.2 Clean Space

Aplicación que mide los niveles de dióxido de carbono del aire que te rodea. En ella puedes ver actualizada la contaminación del aire de tu alrededor pudiendo, además, ver las rutas y lugares más limpios para evitar los sitios con mayor contaminación. Además te advierte de a qué horas y en qué zonas estas cantidades analizadas son más peligrosas para tu salud. [3]



Figura 4: Clean Space

### 2.2.3 ALD Ecodrive

Se trata de una aplicación que evalúa tu estilo de conducción. Mediante pequeños consejos, pretende conseguir que te conviertas en un conductor respetuoso con el medio ambiente.

El seguimiento que realiza de tu conducción mide el consumo de tu combustible y las emisiones de CO<sub>2</sub> derivadas de tu automóvil. [3]



Figura 5: ALD Automotive

## **3 Diseño**

---

### ***3.1 Diseño de la aplicación***

Respecto a lo correspondiente a la sección de diseño de esta aplicación, destacar que no ha supuesto un gran problema. Esto se debe a que forma parte del back-end de un proyecto mayor, por lo que seguro que la parte del front-end llevará consigo un trabajo más exhaustivo en este apartado.

La parte con más peso a la hora de realizar el diseño de la aplicación fue el manejo de los datos, es decir, saber diferenciar entre la que me resultaría realmente útil y la que no, descartando esta última.

A la hora de escoger de entre toda la información almacenada en la base de datos lo primero que me vino a la cabeza fue el resultado que quería alcanzar con el desarrollo de la aplicación y qué necesitaba para lograrlo. De esta manera decidí utilizar de entre toda la información la relacionada con el tráfico, el clima y la contaminación, el resto fue desechado.

Por todo esto y tras realizar un procesado de la información consultada en la base de datos, la aplicación recibe como parámetros datos relacionados con el clima, intensidades de tráfico y niveles de contaminación generando finalmente un resultado tras procesar de nuevo los datos extraídos de la simulación.

### ***3.2 Datasets***

A continuación muestro los formatos en que obtengo la información.

```

{
  "took": 0,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1977,
    "max_score": 1.0,
    "hits": [
      {
        "_index": "vlci_weather",
        "_type": "measurement",
        "_id": "AWIv0BLuGDBC36V5c7Ga",
        "_score": 1.0,
        "_source": {
          "visibility": 10000,
          "main": {
            "humidity": 24,
            "temp": 19.0,
            "temp_min": 19.0,
            "pressure": 1002,
            "temp_max": 19.0
          },
          "clouds": {
            "all": 20
          },
          "sys": {
            "country": "ES",
            "sunrise": 1521180640,
            "sunset": 1521223794,
            "id": 5503,
            "type": 1,
            "message": 0.0031
          },
          "@timestamp": "2018-03-16T14:38:09.023Z",
          "weather": [
            {
              "icon": "02d",
              "description": "few clouds",
              "main": "Clouds",
              "id": 801
            }
          ]
        },
        "@version": "1",
        "id": 2509954,
        "wind": {
          "speed": 7.7,
          "deg": 260,
          "gust": 13.4
        }
      }
    ]
  }
}

```

```

"took": 30,
"timed_out": false,
_shards": {
  "total": 1,
  "successful": 1,
  "skipped": 0,
  "failed": 0
},
"hits": {
  "total": 1551195,
  "max_score": 1.0,
  "hits": [
    {
      "_index": "vlci_intensidad_trafico",
      "_type": "measurement",
      "_id": "AWIo9cgFrIxWP8qTDx5S",
      "_score": 1.0,
      "_source": {
        "@version": "1",
        "features": {
          "properties": {
            "idtramo": "A128",
            "des_tramo": "GUILLEM DE CASTRO",
            "lectura": 1392
          }
        }
      },
      "@timestamp": "2018-03-15T09:20:03.905Z"
    },
    {
      "_index": "vlci_intensidad_trafico",
      "_type": "measurement",
      "_id": "AWIo9cgFrIxWP8qTDx5T",
      "_score": 1.0,
      "_source": {
        "@version": "1",
        "features": {
          "properties": {
            "idtramo": "A13",
            "des_tramo": "ALBORAIA",
            "lectura": 231
          }
        }
      },
      "@timestamp": "2018-03-15T09:20:03.905Z"
    }
  ]
}

```

Figuras 6, 7: Datasets



### 3.3 Diagrama

Como muestra el diagrama, esta aplicación trabaja con datos históricos sobre la contaminación, la intensidad de tráfico y el clima de la ciudad.

Tras ser procesados los datos obtenidos de la base de datos, se obtiene el modelo de día y se lleva a cabo la ejecución de la simulación obteniendo finalmente el resultado final del servicio.

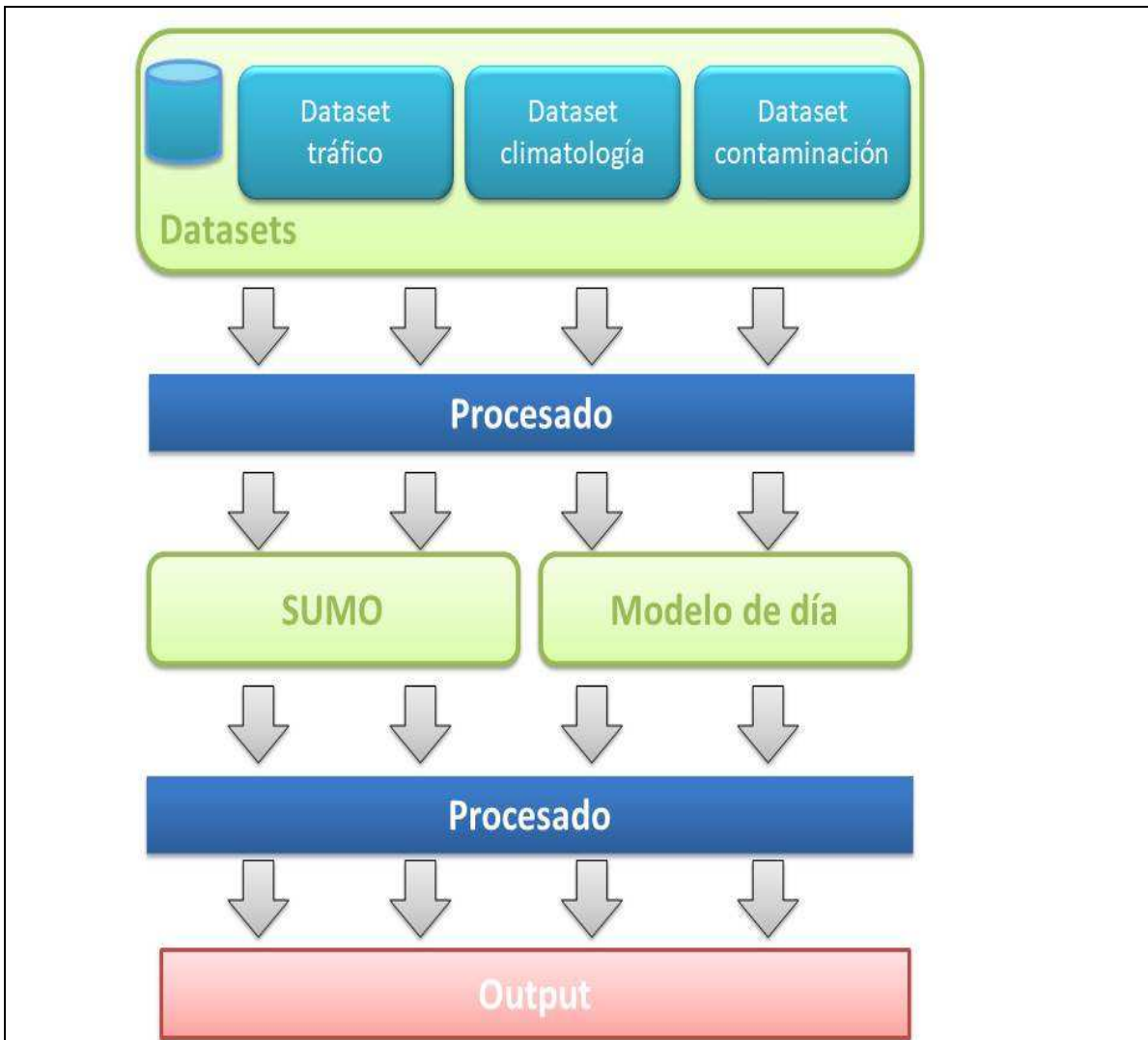


Figura 8: Diagrama de diseño

## 4 Desarrollo

---

### 4.1 Introducción

El desarrollo de este proyecto está suponiendo un reto a nivel personal puesto que, si bien la mayoría de las herramientas que estoy utilizando las conocía de todos estos años de estudio, es la primera vez que mi trabajo es evaluado y utilizado por una empresa en el ámbito laboral.

En cuanto a lo que abarca el desarrollo de la aplicación, el trabajo llevado a cabo se centra en las ciudades españolas de Madrid y Valencia.

### 4.2 Mapas

OpenStreetMap (OSM) es un proyecto colaborativo para crear mapas libres y editables.

Los mapas se crean utilizando información geográfica capturada con dispositivos GPS móviles, ortofotografías y otras fuentes libres. Todos estos datos, tanto las imágenes como los datos vectoriales almacenados en su base de datos, se distribuyen bajo licencia abierta.

El formato de archivo de referencia de OpenStreetMap es el de OSM XML. Cada semana existe una actualización de la base de datos.

OpenStreetMap utiliza una estructura de datos topológica. Los datos primitivos de la cartografía de OSM son:

- Los nodos: puntos que recogen una posición geográfica dada.
- Las vías: lista ordenada de nodos que representa una polilínea o un polígono.
- Las relaciones: grupos de nodos, vías u otras relaciones a las que se pueden asignar propiedades.
- Las etiquetas: se pueden asignar a nodos, caminos o relaciones.

Para la realización de este TFG se ha partido de estos mapas. Básicamente son ficheros en los que se definen las ciudades como si de un grafo se tratara. Estos mapas se caracterizan por utilizar el algoritmo de Dijkstra como algoritmo de enrutamiento.

El motivo por el que se ha elegido este formato se debe a que, como he dicho antes, son accesibles por todo el mundo y en segundo lugar porque SUMO acepta este tipo de archivo para trabajar.

## 4.2.1 Valencia

En lo que a Valencia se refiere es la ciudad que ha sufrido un mayor avance en el desarrollo. Aquí podemos ver el mapa.

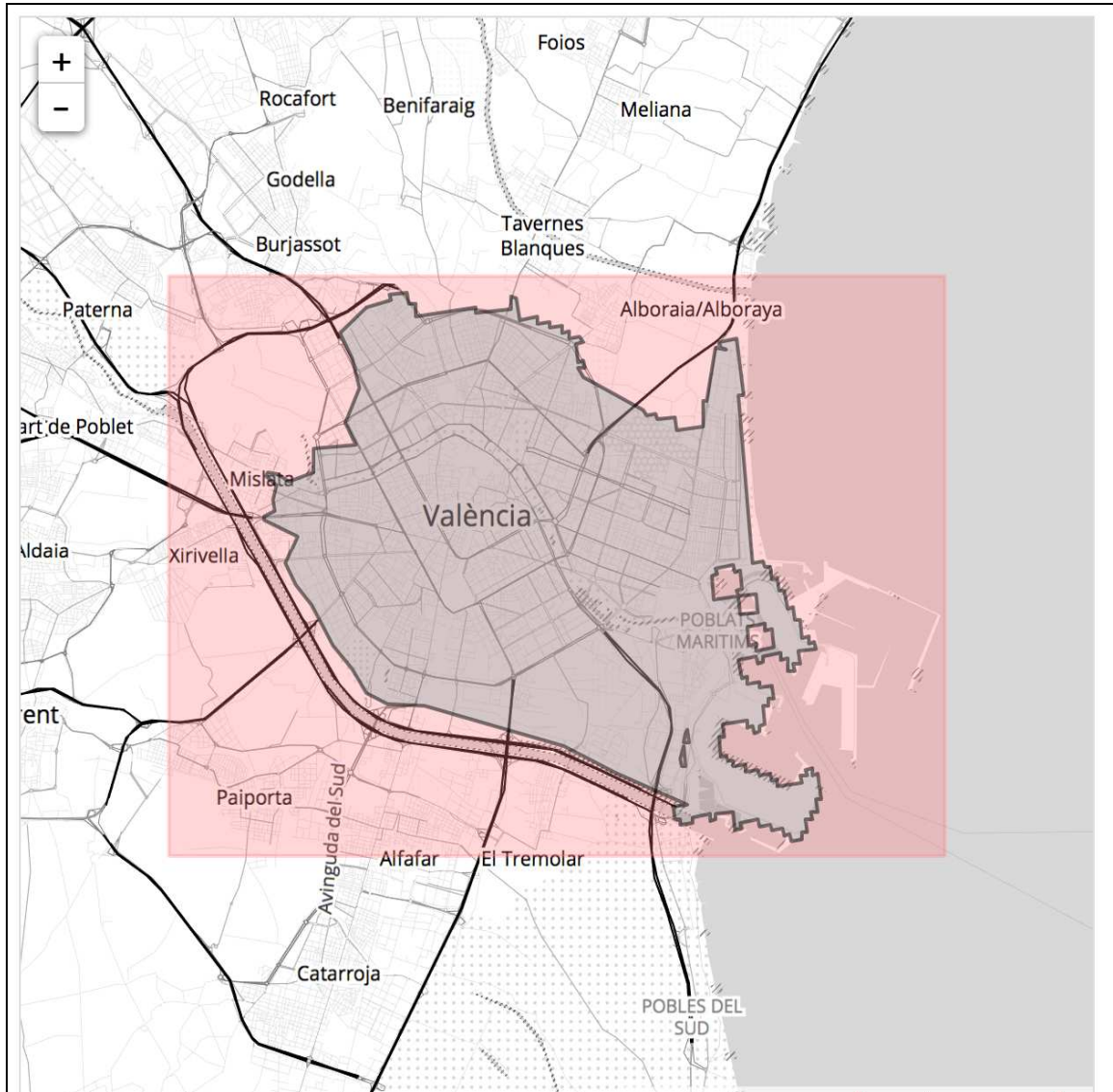


Figura 9: Mapa de Valencia

### 4.2.1.1 Obtención datasets

Respecto a Valencia los datos los han proporcionado desde otro socio del proyecto a través de ElasticSearch, tanto del clima, el tráfico y la contaminación. Accedía a ella a través de una API Rest que me facilitaron.

## 4.2.2 Madrid

Por su parte Madrid ha sufrido más retraso, esto se debe a que desde la empresa dan prioridad a Valencia en estos momentos. A continuación se muestra el mapa de Madrid.

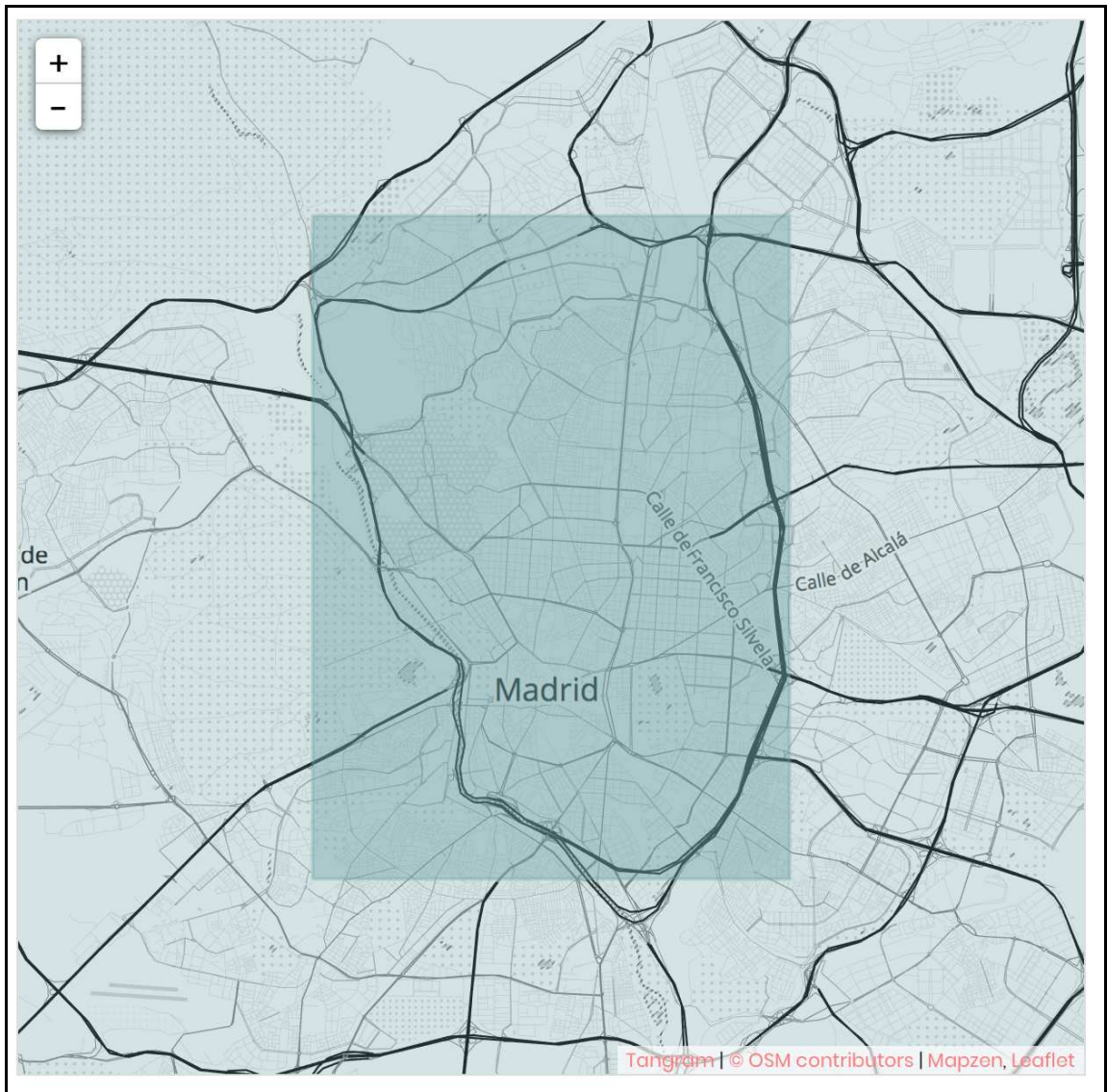


Figura 10: Mapa de Madrid

### 4.2.2.1 Obtención datasets

Respecto a Madrid he obtenido los datos del tráfico y contaminación de la página del ayuntamiento de Madrid y los del clima de Openweather.

Se trata de datos abiertos, lo cual permite obtenerlos a cualquier persona que quiera acceder a ellos. Tienen las siguientes características:

Nombre	Descripción	Descripción en detalle	Frecuencia a actualización	Sector
<b>Datos de calidad del aire: Estaciones de control</b> <a href="#">[4]</a>	Datos diarios del último mes de la estación de contaminación atmosférica.	<p>El Sistema de Vigilancia está formado por 24 estaciones remotas automáticas que recogen la información básica para la vigilancia atmosférica. Poseen los analizadores necesarios para la medida correcta de los niveles de gases y de partículas.</p> <p>Las estaciones remotas son de varios tipos:</p> <ul style="list-style-type: none"> <li>- Urbanas de fondo: Representativas de la exposición de la población urbana en general.</li> <li>- De tráfico: Situadas de tal manera que su nivel de contaminación está influido principalmente por las emisiones procedentes de una calle o carretera próxima, pero se ha de evitar que se midan microambientes muy pequeños en sus proximidades.</li> <li>- Suburbanas: Están situadas a las afueras de la ciudad, en los lugares donde se encuentran los mayores niveles de ozono.</li> </ul>	Mensual	Medio ambiente
<b>Calidad del aire: Datos en tiempo real</b> <a href="#">[4]</a>		El Sistema Integral de la Calidad del Aire del Ayuntamiento de Madrid permite conocer en cada momento los niveles de contaminación atmosférica en el municipio.	Tiempo Real	Medio Ambiente

		<p>En este conjunto de datos puede obtener la información actualizada en tiempo real, actualizándose estos datos cada hora, y esta actualización se hace entre los 20 y los 30.</p>		
<p><b>Tráfico: Intensidad del tráfico, ubicación de los puntos de medida [5]</b></p>	<ul style="list-style-type: none"> <li>- IDelem</li> <li>- Tipo_elem</li> <li>- Identif</li> <li>- Nombre</li> <li>- St_x</li> <li>- St_y</li> </ul>	<p>Tráfico: intensidad del tráfico, puntos de medida. Este conjunto de datos está relacionado con el de "tráfico: intensidad del tráfico, valores", indicando este último para cada punto de medida, los vehículos que pasan:</p> <ul style="list-style-type: none"> <li>- La infraestructura de puntos de medida, disponible en la ciudad de Madrid se corresponde con: 7.360 detectores de vehículos: disponiendo en 71 de ellos de dispositivos de lectura de matrículas, en 158 de ellos de sistemas ópticos de visión artificial con control desde el Centro de Gestión de Movilidad, 1.245 específicos de vías rápidas y acceso a la ciudad y el resto de los 5.886 sistemas básicos de control de semáforos.</li> <li>- Más de 3.600 puntos de medida: disponiendo en 253 de ellos de los sistemas para el control de velocidad, caracterización de los vehículos y doble lazo de lectura y 70 de ellos conforman las estaciones de toma de aforos específicas de la</li> </ul>	<p>Tiempo real</p>	<p>Transporte</p>

		<p>ciudad.</p> <ul style="list-style-type: none"> <li>- Sistemas de control automático de toda la información obtenida de los detectores con el contraste continuo con patrones de comportamiento esperado, así como el seguimiento de las instrucciones marcadas por el Comité Técnico de Normalización AEN/CTN 199 “Equipamiento para la Gestión del Tráfico” y en particular las aplicaciones específicas del SC3 relativas a “Detectores y estaciones de toma de datos” y SC15 relativas a “Calidad de datos”.</li> </ul>		
<p><b>Datos climatológicos</b> <a href="#">[4]</a></p>	<ul style="list-style-type: none"> <li>- Temp. Max</li> <li>- Temp. Min</li> <li>- Humedad.</li> <li>- Descripción.</li> <li>- Id. Estación</li> </ul>	<p>Medición de distintos parámetros mediante el uso de estaciones colocadas en distintos puntos de la ciudad.</p>	<p>Tiempo real</p>	<p>Meteorología</p>

Tabla 1: Open Data Madrid.

La obtención de la información relacionada con el tráfico y el clima de Madrid la he logrado mediante un script. Este programa se encarga de descargar los open data correspondientes a la intensidad de tráfico de la página del ayuntamiento de Madrid y de la página OpenWeather lo relativo a la meteorología y contaminación. Los datos del clima y contaminación se descargan en formato JSON mientras que los del tráfico en XML.

Una vez descargados ambos documentos, realizaba un procesado de los mismos para obtener los que realmente me interesaban, puesto que del tráfico llega información para prácticamente todas las calles de Madrid y respecto del clima de muchas estaciones meteorológicas.

Una vez filtrada la información la convertía en formato JSON (los que hacen falta) almacenándola por último en MongoDB. De todo el procesado y almacenado de datos se encarga un programa desarrollado en Python, el cual es llamado por el script.

El script es lanzado cada tres horas todos los días gracias a Crontab.

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 */3 * * * cd /home/dev/Documents/ESTABLISH/Intensidad_trafico_Madrid && ./download_traffic.sh >> /home/dev/Documents/ESTABLISH/Intensidad_trafico_Madrid/cron_traffic.log
```

Figura 11: Fichero Crontab.

Por otra parte obtuve información menos específica respecto al clima y al tráfico en estas mismas páginas de años anteriores, descargándola por mi cuenta e introduciéndola en la base de datos mediante consultas.

## 4.3 Infraestructura de desarrollo

En lo que a equipo se refiere utilizado para el desarrollo del servicio, tiene las siguientes características.

### 4.3.1 Hardware

El hardware utilizado para este proyecto ha sido el proporcionado por mi empresa, un HP ProBook 4520s con un procesador Intel Core i3 CPU M 380 con 4 núcleos a 2.53GHz, 4 GB de memoria RAM y un disco duro de 150 GB.

También he necesitado conectarme al servidor de la empresa para realizar las simulaciones puesto que eran demasiado pesadas para realizarlas en el ordenador. Cuenta con 8GB de memoria principal y un procesador Intel Xeon CPU E5-2690 v2 @ 3.00GHz.

### 4.3.2 Software

Respecto al software se ha utilizado como sistema operativo Linux, en su distribución Ubuntu con versión 16.04 LTS.

El servidor cuenta con Ubuntu en su versión 16.



## **4.4 Lenguajes de programación**

He necesitado utilizar varios lenguajes de programación donde cada uno cumpliera una función. La mayoría de ellos los conocía pero debo decir que he tenido que profundizar en ellos puesto que los había utilizado para tareas sencillas. En el caso de otros he tenido que empezar desde cero documentándome para efectuar mi aprendizaje.

### **4.4.1 Python**

Lenguaje de programación multiparadigma, que soporta orientación a objetos, programación imperativa y programación funcional. Es interpretado y multiplataforma.

Para desarrollar los módulos utilizados en el proyecto me he decantado por Python. Se trata de un lenguaje de programación interpretado y multiplataforma.

Fue la empresa quien me indico que tenía que utilizar este lenguaje de programación. Si de mi hubiese dependido la verdad que habría sido una de mis alternativas, puesto que me resultaba familiar de asignaturas como Diseño y análisis de algoritmos o Fundamentos de aprendizaje automático, pero sobre todo por la necesidad de trabajar con Elasticsearch y MongoDB, puesto que ambas admiten este lenguaje de programación con la inclusión de librerías que facilitan mucho el trabajo. También es sencilla la interacción con documentos XML y JSON. Una buena alternativa es Java, los motivos que argumento son los mismos y además, en mi caso, me es muy familiar por asignaturas como Análisis y diseño del software.

### **4.4.2 Scripting**

Lenguaje normalmente interpretado. Llamado lenguaje de guion.

Decir que tenía nociones básicas de este lenguaje adquirido en mi paso por la universidad, pero desconocía su potencial. Me ha ahorrado muchísimo tiempo pudiendo automatizar tareas. Una vez lanzaba el script no tenía que preocuparme de ir ejecutando aplicación a aplicación individualmente. También me ha sido útil cada vez que descargaba información para almacenar en la base de datos de Madrid.

La decisión de utilizar este lenguaje fue mía. Decidí decantarme por este lenguaje porque lo conocía ligeramente de la carrera, ya que lo vi en Arquitectura de computadores, resultándome realmente útil para lanzar distintas aplicaciones y programas realizados por mí. De ahí proviene esta decisión, de la necesidad de tener que lanzar una serie de programas secuencialmente despreocupándome de ir una a una.

### **4.4.3 Query DSL**

Lenguaje de dominio específico utilizado para realizar consultas a los documentos indexados en Elasticsearch.

Nunca había trabajado con este tipo de consultas puesto que hasta este momento conocía SQL para bases de datos relacionales. Necesité un tiempo de aprendizaje pero existe bastante documentación en internet que facilitó este proceso.

## **4.5 Herramientas de desarrollo**

Para la realización de este proyecto he requerido de diversas herramientas. Aunque muchos de los conceptos necesitados para su utilización ya los tenía adquiridos, he tenido que formarme para ser capaz de dominar otras muchas.

A continuación procedo a explicar las herramientas de las que me he ayudado.

### **4.5.1 Elasticsearch**

Es un motor de búsqueda orientado a documentos que nos permite indexar un gran volumen de datos para poder realizar consultas posteriormente. Guarda los documentos en formato JSON y es no relacional. Una de las principales diferencias con las bases de datos relacionales vistas a lo largo de la carrera es que no necesita seguir un esquema, pudiendo tener documentos de una misma colección (concepto similar a una tabla relacional) con diferentes esquemas.

En este caso se ha utilizado para almacenar los datos históricos sobre intensidad de tráfico en diferentes momentos del día así como de la climatología. Por otra parte decir que tuve que realizar una importante tarea de aprendizaje, puesto que al contrario de las vistas en la universidad, se trata de bases de datos no relacionales.

Respecto a esta elección no tuve nada que ver, se trataba de una decisión determinada por otro miembro del consorcio europeo que eran los encargados de su desarrollo, en nuestro caso solo nos ha sido necesario realizar consultas sobre ella por lo que no se ha requerido desarrollo por nuestra parte.

Atravesé un importante periodo de documentación para aprender a manejarme bien con Elasticsearch. [\[6\]](#)

En mi opinión es una buena decisión puesto a que permite almacenar documentos sin una estructura fija (al contrario que las bases de datos relacionales), lo cual viene muy bien debido a que existen distintos tipos de documento con atributos diferentes en algunos casos. Si de mí dependiera habría dudado entre Elasticsearch y MongoDB, puesto que son muy similares y permiten almacenar documentos de manera indexada sin preocuparte de la estructura. Es por esto que me parece una decisión correcta.

## 4.5.2 MongoDB

Se trata de una base de datos que al igual que Elasticsearch es no relacional. Es orientada a documentos, por lo que los datos no son guardados en registros. Estos documentos son almacenados en formato JSON

En mi caso he utilizado MongoDB para almacenar los datos sobre el tráfico y el clima de Madrid (la contaminación es un punto que quedó pendiente). Se trataba de una nueva tecnología para mí por lo que tuve que realizar un estudio sobre ella. [\[7\]](#)

En este caso la elección fue cosa de la empresa para la que he trabajado. Mi razonamiento es el mismo que en el caso de Elasticsearch, puesto que los documentos almacenados son prácticamente iguales. Mi alternativa sería Elasticsearch, pero al igual que antes, me parece una buena decisión.

## 4.5.3 Robo3T

Interfaz gráfica para poder interactuar con bases de datos MongoDB.

De gran utilidad a la hora de realizar consultas a la base de datos de Madrid. [\[8\]](#)

## 4.5.4 API Rest

Una API Rest es un servicio que nos provee de funciones que nos dan la capacidad de hacer uso de un servicio web que no es nuestro, dentro de una aplicación propia, de manera segura.

En este caso la he utilizado para acceder a los servicios proporcionados por Valencia mediante consultas. Utilicé esta API. [\[9\]](#)

## 4.5.5 Cron

Cron es un administrador regular de procesos en segundo plano (demonio) que ejecuta procesos a intervalos regulares. Se ayuda de Crontab para dicha función.

No conocía la existencia de este tipo de herramientas puesto que en el grado no se ve nada al respecto. Destacar que gracias a este administrador he podido obtener datos fiables de todos los días a la misma hora, pudiendo sacar patrones respecto a esos momentos. De haberlo tenido que hacer manualmente habría perdido mucho tiempo, además es seguro que se me hubiese pasado por alto en más de una ocasión.

### 4.5.5.1 Crontab

Fichero de texto donde se especifica una lista de comandos a ejecutar en un tiempo especificado por el usuario.

Con este fichero indiqué que se ejecutase un script con el cual descargaba los open data sobre el clima y el tráfico en Madrid y los procesaba, para por ultimo almacenarlos en la base de datos de Madrid. Esta acción se ejecutaba cada tres horas. [\[10\]](#)

## 4.5.6 Rdesktop

Cliente de protocolo de escritorio remoto, es una aplicación cliente que permite controlar la sesión de escritorio en otro sistema que esté ejecutando un servidor compatible.

La utilización de esta herramienta ha sido vital, puesto que las simulaciones las he ejecutado en el servidor de la empresa, conectándome a través de Rdesktop. De no ser así no habría podido efectuarlas ya que mi equipo no lo soportaba.

## 4.5.7 SUMO (Simulation of Urban MOvility)

Herramienta central para el desarrollo de este proyecto. Se trata de una aplicación para la simulación de tráfico publicada como código abierto que permite la personalización de muchos parámetros dando como resultado la contaminación generada, ruta o distancia recorrida de los distintos vehículos que conforman la simulación.

Destacar el importantísimo proceso de documentación que he tenido que llevar a cabo de la aplicación puesto que proporciona muchísimas alternativas. [\[11\]](#)

SUMO trabaja exclusivamente con ficheros XML tanto de entrada como en la generación de los resultados. Las calles que forman la ciudad son nodos identificados de manera única mediante un id.

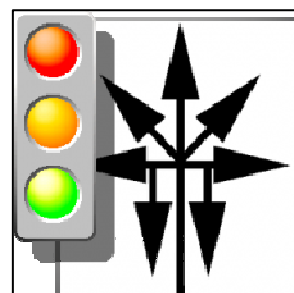


Figura 12: Logo SUMO.

Hay que decir que los pilares de SUMO son las herramientas RandomTrips, Duarouter y Sumo-Gui.

### 4.5.7.1 *RandomTrips*

Utilizada para crear los automóviles que formarán parte de la simulación. Genera un fichero XML con el origen y el destino de todos los vehículos.

### 4.5.7.2 *Duarouter*

A raíz del fichero generado por RandomTrips, crea uno nuevo especificando toda la ruta que ha de seguir el transporte.

### 4.5.7.3 Sumo-Gui

Interfaz gráfica para visualizar la simulación. Muestra visualmente la simulación en tiempo real pudiendo apreciar atascos, flujos que siguen los coches...

## 4.5.8 Google Maps

Es un servidor de aplicaciones de mapas en la web que pertenece a Alphabet Inc (cuya principal subsidiaria es Google). Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo y rutas entre distintas ubicaciones.

Se trata de otra de las principales herramientas sobre las que se apoya este proyecto. Gracias a Google Maps he sido capaz de realizar un estudio exhaustivo acerca de los flujos de tráfico que se producen en la ciudad. Además ofrece un servicio que te muestra la congestión de tráfico producido cualquier jornada de la semana, en el lugar y la hora del día seleccionada.

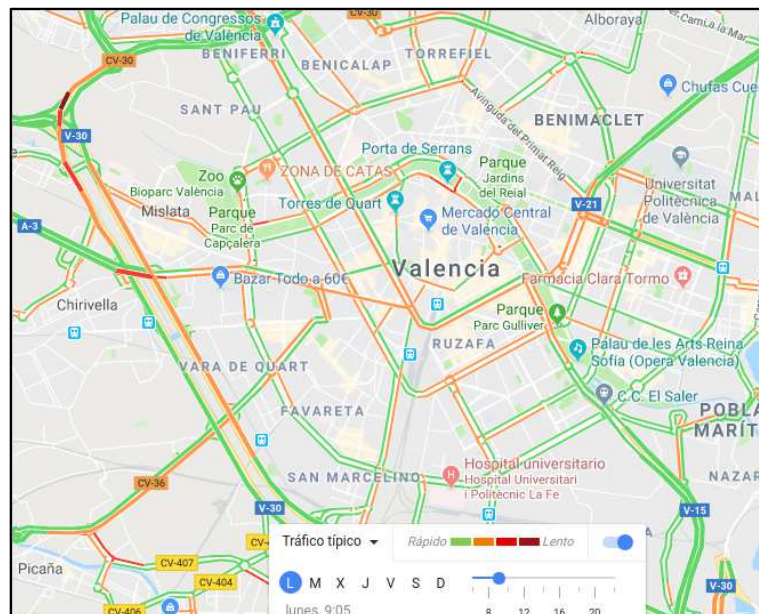


Figura 13: Google Maps.

El uso de esta herramienta fue decisión mía. Dicha decisión surgió cuando me encontré el problema de que las rutas no tenían ningún sentido, ya que como he explicado antes estos mapas se apoyan en Dijkstra como algoritmo de enrutamiento, por lo que los pesos asignados a los caminos no se corresponden con los que deberían tener en la realidad. Por este motivo los atascos surgían en calles con las que no se corresponden en la vida real y por consiguiente la contaminación.

Ante el nacimiento de dicho problema busqué alternativas para resolverlo. Así fue como llegué a la conclusión de que si conseguía unos flujos realistas de tráfico los atascos se crearían dónde deben con su correspondiente contaminación. De esta manera solucioné el problema. [\[12\]](#)

## 4.6 Estudio del flujo de la circulación

En primer lugar debo destacar la importancia de este punto. Considero que ha sido la parte más costosa del proyecto y así ha quedado reflejado en tiempo dedicado, habiendo superado el mes.

Al efectuar simulaciones en Sumo y visionarlas me encontré con un serio problema, las rutas que tomaban los vehículos para llegar a su destino no tenían nada que ver con el que una persona seguiría en un caso real. Además, en la simulación por defecto no podía controlar el hecho de que los flujos tuviesen sentido. Por ejemplo en Madrid, por las mañanas y en hora punta existe un importante flujo de entrada desde el sur (Alcorcón, Móstoles, Getafe...). Sin embargo al generar de manera aleatoria el tráfico con SUMO comportamientos como este no quedaban reflejados.

Esto me suponía un obstáculo para lograr mi objetivo, que no era otro que conseguir asemejar los valores de la simulación a los proporcionados por los dataset, puesto que carreteras tan importantes como la m-30 de Madrid no tenían prácticamente circulación alguna, obteniendo niveles de contaminación en dichos puntos irrisorios.

Para solucionar el problema decidí dividir el mapa en distintas regiones, asociando a cada una un id que la diferenciara de manera única.

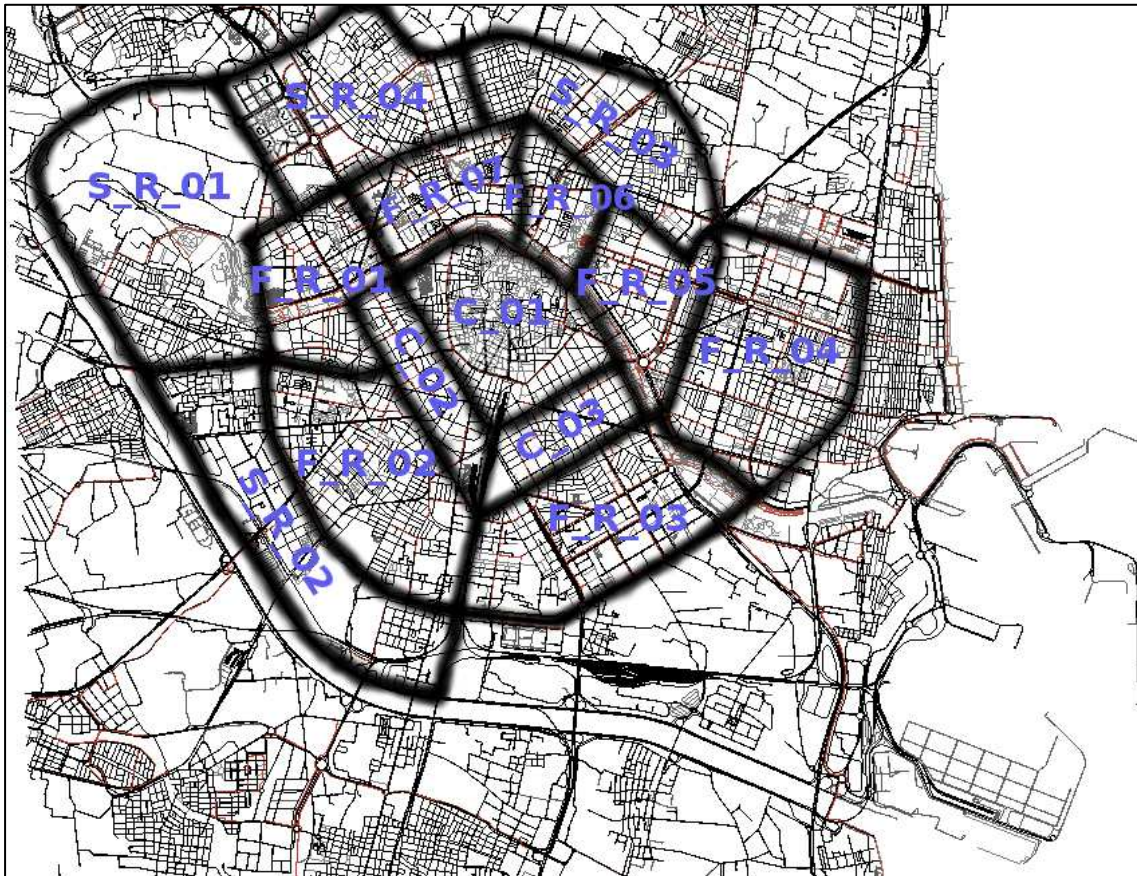


Figura 14: Distritos de Valencia.

Una vez dividida la ciudad en distritos, tenía que reflejarlo en la aplicación SUMO. Para ello accedí al mapa mediante la herramienta de visualización Sumo-Gui y comencé a definir las zonas añadiendo una gran cantidad de calles mediante sus ids.

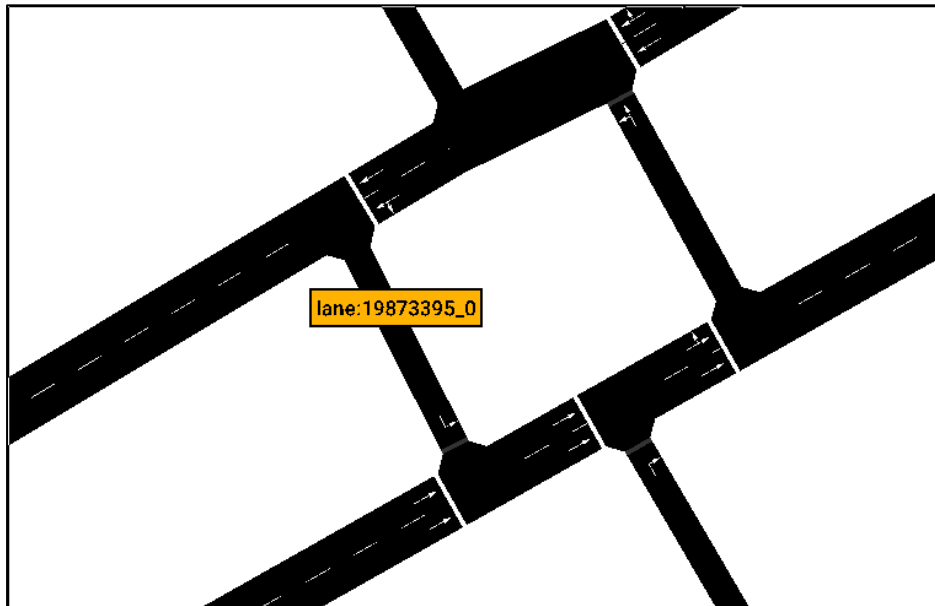


Figura 15: ID de las calles.

Tras obtener los distintos distritos era hora de estudiar el trayecto que efectuarían los automóviles entre cada dos puntos. En este instante me fue extremadamente útil la aplicación Google Maps. Seleccionando un punto origen y otro destino esta herramienta muestra cual es el camino más óptimo en cuanto a tiempo de trayecto se refiere que es el criterio que he utilizado.

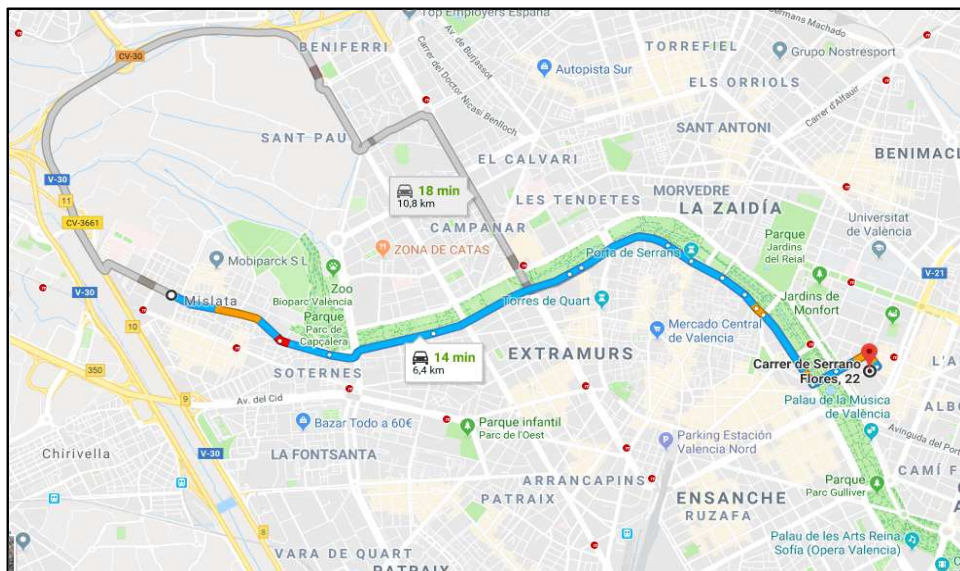


Figura 16: Ruta óptima

Para reflejar estos flujos procedí a desarrollar un nuevo módulo Python. Como he explicado antes SUMO trabaja con archivos XML, generando un fichero en el que se indica el origen y destino de cada trayecto, pero también proporciona herramientas de personalización permitiendo indicar puntos que tiene que atravesar.

De esta manera el programa contempla todos los posibles viajes entre distritos, modificando los orígenes y destinos y añadiendo las calles intermedias necesarias al fichero. Para añadir los puntos intermedios consulté con Sumo-Gui los ids de las calles que me interesaban, obligando a los coches a pasar por ahí y plasmando así el flujo real en SUMO. Decir que en ocasiones no solo había un único camino valido, en esos casos el programa contempla todas las rutas asignándole un porcentaje de posible generación a cada una en función de su importancia.

Además con este programa controlo que porcentaje del tráfico total quiero que vaya de una zona a otra.

#### **4.7 Colocación de las estaciones de tráfico**

SUMO permite controlar la intensidad de tráfico existente en cada carretera mediante la colocación de estaciones. Estas estaciones devuelven el número total de vehículos que pasan por ellas, así como la intensidad del tráfico medio en un intervalo de tiempo dado, medido en vehículos/hora.

La decisión de la ubicación de las estaciones no ha sido al azar. Me he fijado en las calles con mayor densidad de tráfico en cada ciudad a través de consultas a la base de datos, situándolas ahí. Para ello me he creado un fichero .xml de configuración en el que indico los ids de los carriles donde quiero situar las estaciones de medición. La obtención de los ids la he llevado a cabo mediante Sumo-Gui. Al finalizar la simulación se genera otro fichero .xml donde se muestran las intensidades de tráfico de cada carril.

Para hacer que los resultados sean legibles he realizado otro programa que se encarga de leer el fichero .xml resultante sumando las intensidades de los distintos carriles en un mismo punto (conociendo así la intensidad total en la carretera) y mostrándolas por zonas.

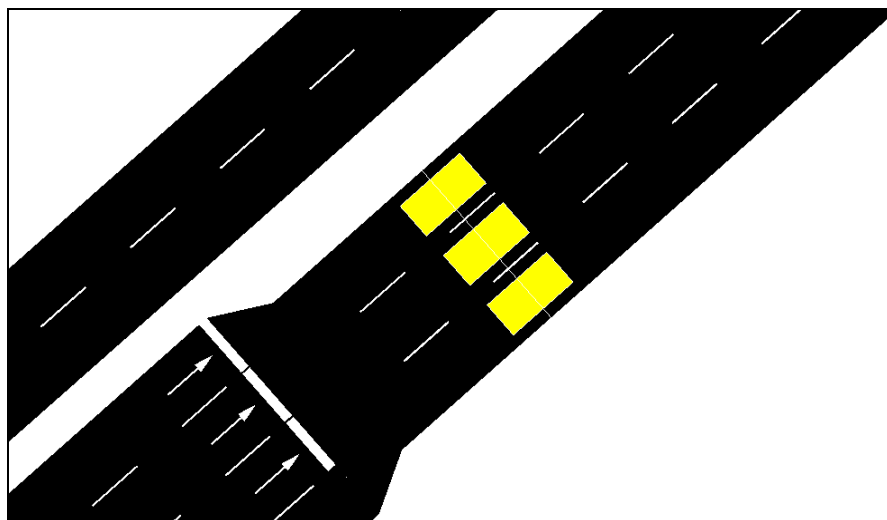


Figura 17: Estaciones de tráfico.



## **4.8 Colocación de las estaciones de contaminación**

Al igual que con el tráfico, he tenido que colocar estaciones para obtener los niveles de contaminación. Para ello, también he realizado consultas a la base de datos para ver donde se ubicaban las estaciones reales, de esta manera sabía en qué punto del mapa me tenía que fijar.

Para realizar esta tarea he tenido que modificar el fichero de configuración, una vez cambiado la simulación me mostraba todas las calles del mapa con sus respectivos niveles. Para facilitar la tarea he realizado un programa que realiza dicha lectura y la muestra por pantalla.

## **4.9 Obtención de los diferentes modelos de día**

Uno de los factores que más influye en la producción de tráfico y contaminación de vehículos es sin duda la meteorología, puesto que en un día lluvioso es casi seguro que existirá mayor densidad de la circulación que en una jornada soleada donde el tránsito será más fluido. Lo mismo ocurre al diferenciar entre días laborables y fines de semana, puesto que los comportamientos en ciertas horas de la jornada son completamente distintos entre, por ejemplo, un lunes y un domingo. Todos estos factores afectan en la producción final de contaminación generada por automóviles sobre la ciudad.

Teniendo todo esto en cuenta hay que decir que uno de los parámetros más importantes que contempla este proyecto para las simulaciones es el tipo de día. Para obtener dicho modelo me he ayudado de un nuevo programa desarrollado en Python. Este programa analiza en un comienzo si se trata de una jornada laboral o no (en realidad hace una comprobación de si está situado entre el lunes y el viernes, pero en caso de que salga un valor en cuanto a intensidad de tráfico excesivamente bajo será descartado por ser considerado festivo).

Las fechas que pasan este primer filtro acceden a examinarse en otro. En dicho segundo filtro se efectúa una consulta con la fecha señalada para obtener su climatología entre dos horas (normalmente las realizaba fijándome en tres valores espaciados en el tiempo obteniendo así una mayor fiabilidad). Los posibles valores sobre el clima traen asociados un id individualmente, para saber si cada uno de los días cumplía o no el requisito consulté en la base de datos los ids de las condiciones meteorológicas que me interesaban para el modelo en particular y simplemente realizaba una comparación entre este último id y el correspondiente a la jornada tratada. De nuevo, de no cumplir este requisito la fecha era descartada.

La parte final del programa hace una última consulta entre dos horas del día sobre la base de datos del tráfico con las jornadas que sobreviven a los dos filtros anteriores. Dicha consulta retorna todas las calles de la ciudad registradas con su respectiva intensidad de tráfico, de manera que coge las de las calles que me interesan y tras un procesado efectuado junto al resto de intensidades de los otros días válidos, obtiene la intensidad media de tráfico calculada en vehículos/hora.

Por último muestra por pantalla los resultados logrados, de esta forma conozco los valores que tienen que alcanzar las estaciones de tráfico de SUMO para lograr este modelo de jornada.

```
----Datos para modelo de dia laborable y cielo despejado:----  
-----  
Intensidad de trafico en A82(Acceso norte) es de: 6797(v/h)  
Intensidad de trafico en A373(Acceso sur) es de: 7788(v/h)  
Intensidad de trafico en A30(Acceso sureste) es de: 2848(v/h)  
Intensidad de trafico en A72(Acceso oeste) es de: 7168(v/h)  
Intensidad de trafico en A1(Acceso este) es de: 7325(v/h)  
Intensidad de trafico en A59(Acceso suroeste) es de: 6427(v/h)  
miguel@miguel-HP-ProBook-4520s:~/Escritorio/Establish/Mapas/Valencia$
```

Figura 18: Modelo de día.

## 5 Integración, pruebas y resultados

En este apartado muestro la integración de los diversos módulos desarrollados así como los resultados de las simulaciones efectuadas. Destacar que uno de los principales motivos por el que no he completado el desarrollo total del servicio ha sido el tiempo de ejecución.

Estas simulaciones las he llevado a cabo mediante el servidor de la empresa. Aun así a poco que metiera cantidad de tráfico para lograr equiparar los resultados en los accesos de la ciudad a la realidad, se me podía ir por encima de las 5 horas perfectamente, pudiendo lanzar un máximo de 4 simulaciones a la vez.

El otro motivo por el que no he llegado a terminar la aplicación se debe a que en el primer mes de desarrollo el trabajo se enfocó en Madrid, puesto que no teníamos los datasets de Valencia. Pasado este periodo Valencia se convirtió en prioritario, teniendo que dejar de lado lo realizado hasta ese punto, comenzando de nuevo desde cero.

```
-----NORTE-----
Intensidad de trafico media total: 1024.95 v/h.
El numero total de vehiculos de entrada es: 582 v.
Entrada norte (intensidad media de trafico): 583.05 v/h.
El numero total de vehiculos de salida es: 437 v.
Salida norte (intensidad media de trafico): 441.9 v/h.
-----OESTE-----
Intensidad de trafico media total: 2108.57 v/h.
El numero total de vehiculos de entrada es: 1073 v.
Entrada oeste (intensidad media de trafico): 1075.45 v/h.
El numero total de vehiculos de salida es: 1017 v.
Salida oeste (intensidad media de trafico): 1033.12 v/h.
-----SUROESTE-----
Intensidad de trafico media total: 1729.26 v/h.
El numero total de vehiculos de entrada es: 856 v.
Entrada suroeste (intensidad media de trafico): 860.55 v/h.
El numero total de vehiculos de salida es: 854 v.
Salida suroeste (intensidad media de trafico): 868.71 v/h.
-----SUR-----
Intensidad de trafico media total: 1783.17 v/h.
El numero total de vehiculos de entrada es: 973 v/h.
Entrada sur (intensidad media de trafico): 973.7 v/h.
El numero total de vehiculos de salida es: 786 v/h.
Salida sur (intensidad media de trafico): 809.47 v/h.
-----SURESTE-----
Intensidad de trafico media total: 1192.46 v/h.
El numero total de vehiculos de entrada es: 651 v/h.
Entrada sureste (intensidad media de trafico): 655.2 v/h.
El numero total de vehiculos de salida es: 525 v/h.
Salida sureste (intensidad media de trafico): 537.26 v/h.
-----NORESTE-----
Intensidad de trafico media total: 1748.99 v/h.
El numero total de vehiculos de entrada es: 913 v.
Entrada noreste (intensidad media de trafico): 917.55 v/h.
El numero total de vehiculos de salida es: 788 v.
Salida noreste (intensidad media de trafico): 831.44 v/h.
```

Figura 19: Salida del programa.

Cómo se puede ver en la anterior imagen, me he centrado en los accesos a Valencia. En un principio me surgió el problema de no saber por qué zona debía empezar, para

decidirme me fijé en cuales eran las calles con mayor intensidad de tráfico pues serían los puntos donde se encontrarían los niveles más altos de contaminación.



Figura 20: Intensidad de tráfico en Valencia.

En la imagen se aprecia cómo las franjas negras y azul marino (mayores niveles de intensidad media de tráfico) se ubican en los accesos. De esta forma me centré en estos puntos como punto de partida intentando alcanzar esas intensidades.

El primer modelo que contemplé fue el de jornada laborable y despejada en hora punta, aplicando como criterio para la hora punta de 7 a 9 de la mañana. Es por esto por lo que todas las simulaciones contemplan un periodo de dos horas (esto no significa que tardasen ese tiempo, puesto que era muy superior).

Los datos a lograr por tanto fueron:

Tramo	Intensidad media total (v/h)
Norte	6797
Oeste	7168
Suroeste	6427
Sur	7788
Sureste	2848
Noreste	7325

Tabla 2: Datos objetivo.

Al principio creí que esta tarea sería sencilla, puesto que mi primera idea fue utilizar el modulo con que direccionaba los flujos aplicándole un porcentaje del tráfico total en los puntos deseados, es decir, si por ejemplo generaba una simulación de una hora (3600 segundos) con un vehículo creado cada segundo y aplicaba un 10% del tráfico como

entrada en un acceso y otro 10% de salida obtendría una intensidad de 720v/h. Si bien en este sencillo caso así era, en general no podía estar más equivocado.

Cómo en un primer instante no sabía si el comportamiento sería el esperado comencé realizando simulaciones con valores aleatorios para comprobarlo. Decir que he efectuado muchas más simulaciones de las que aquí muestro, pero estas son las más representativas.

Simulación con valores por defecto:

Tráfico generado: 7200 vehículos (1 vehículo/seg).

Tráfico direccionado: 0 vehículos

Acceso	% Entrada	Intensidad media(v/h)	% Salida	Intensidad media(v/h)	Int-total media(v/h)
Norte	Por defecto	61.28	Por defecto	82.56	143.84
Oeste	Por defecto	252.12	Por defecto	250.09	502.21
Suroeste	Por defecto	198.84	Por defecto	319.81	518.65
Sur	Por defecto	0	Por defecto	0	0
Sureste	Por defecto	51	Por defecto	158.68	209.68
Noreste	Por defecto	0	Por defecto	6.28	6.28

Tabla 3: Simulación por defecto.

Simulación 1:

Tráfico generado: 7200 vehículos (1 vehículo/seg).

Tráfico direccionado: 47%, 3384 vehículos.

Acceso	% Entrada	Intensidad media(v/h)	% Salida	Intensidad media(v/h)	Int-total media(v/h)
Norte	3,9	202.91	3,9	181.83	384.74
Oeste	3,9	445.37	3,9	516.66	962.03
Suroeste	3,9	360.19	3,9	412.1	772.29
Sur	3,9	294.92	3,9	360.57	655.49
Sureste	3,9	264.28	3,9	240.64	504.92
Noreste	3,9	301.82	3,9	335.92	637.74

Tabla 4: Simulación 1.

Simulación 2:

Tráfico generado: 14400 vehículos (1 vehículo/0.5 seg).

Tráfico direccionado: 47%, 3384 vehículos.

Acceso	% Entrada	Intensidad media(v/h)	% Salida	Intensidad media(v/h)	Int-total media(v/h)
Norte	3,9	113.53	3,9	94.94	208.47
Oeste	3,9	229.59	3,9	247.91	477.5
Suroeste	3,9	189.03	3,9	219.17	408.2
Sur	3,9	155.5	3,9	155.5	311.0
Sureste	3,9	130.16	3,9	118.04	248.2
Noreste	3,9	156.07	3,9	158.32	314.39

Tabla 5: Simulación 2.

Simulación 3:

Tráfico generado: 24000(1 vehículo/0.3 seg)

Trafico direccionado: 47%, 11280 vehículos

Acceso	% Entrada	Intensidad media(v/h)	% Salida	Intensidad media(v/h)	Int-total media(v/h)
Norte	3,9	113.17	3,9	100.59	213.76
Oeste	3,9	226.16	3,9	260.1	486.26
Suroeste	3,9	191.72	3,9	219.1	410.82
Sur	3,9	161.57	3,9	161.23	322.8
Sureste	3,9	134.87	3,9	118.33	253.2
Noreste	3,9	162.09	3,9	162.95	325.04

Tabla 6: Simulación 3.

Simulación 4:

Tráfico generado: 9000 (vehículo/0.8 seg)

Trafico direccionado: 75%, 6750 vehículos

Acceso	% Entrada	Intensidad media(v/h)	% Salida	Intensidad media(v/h)	Int-total media(v/h)
Norte	6,9	181.52	5.5	123.29	304.81
Oeste	6,9	309.37	5.5	290.25	599.62
Suroeste	6,9	259.78	5.5	242.03	501.81
Sur	6,9	282.52	5.5	225.2	507.72
Sureste	6,9	190.62	5.5	146.95	337.57
Noreste	6,9	282.52	5.5	226.56	509.08

Tabla 7: Simulación 4.

### Simulación 5:

Tráfico generado: 36000 (1 vehículo/0.2 seg)

Trafico direccionado: 75%, 27000 vehículos

Acceso	% Entrada	Intensidad media(v/h)	% Salida	Intensidad media(v/h)	Int-total media(v/h)
Norte	6.9	201.39	5.5	137.67	339.06
Oeste	6.9	343.32	5.5	319.2	662.52
Suroeste	6.9	289.47	5.5	264.48	553.95
Sur	6.9	302.96	5.5	247.36	550.32
Sureste	6.9	211.88	5.5	155.91	367.79
Noreste	6.9	310.33	5.5	248.37	558.7

Tabla 8: Simulación 5.

Podemos ver como los resultados obtenidos son muy bajos.

La aplicación SUMO genera prácticamente todo el tráfico en un único carril, de manera que por ejemplo, en autopistas de 3 carriles, desaprovechaba casi  $\frac{2}{3}$  partes del tráfico del que se generaría en condiciones normales.

Tras no cumplir mis expectativas decidí que debía seguir documentándome sobre la herramienta. De forma que encontré información acerca de la configuración, probé a cambiar el modelo de circulación (por defecto utiliza Krauss y lo modifiqué a IDM o conducción inteligente), disminuí la distancia entre vehículos...además de todas estas di con una opción que me permitía definir un flujo de un punto a otro especificando la intensidad de tráfico (v/h) deseada. Supuse que esta sería la solución a mi problema, pero en absoluto...los resultados continuaban siendo bajísimos. Resulta que SUMO funciona bien cuando tiene que generar un vehículo por segundo (o un tiempo mayor), pero cuando le indicas que lo haga en intervalos menores su rendimiento no es el deseado.

Tras esto decidí acotar las simulaciones a un único acceso para comprender que sucedía.

Tras aplicar los cambios de configuración comentados comencé a ver mejoras significativas. La mayor intensidad de tráfico que alcancé fue de unos 7000 v/h, todavía lejos de los 8000 que necesito.

Tramo	Intensidad media total (v/h)
Norte	6752
Oeste	6844
Suroeste	6402
Sur	6859
Sureste	2849
Noreste	6806

Tabla 9: Simulación 6.

Sumo me limitaba los resultados, impidiéndome que fueran más cercanos a la realidad. Es por ello que tuve que buscar una solución, tomando la decisión de que la mejor opción era extrapolar los datos logrados a los reales, es decir, partiendo de los datos reales de contaminación e intensidad de tráfico con los que contábamos y de los obtenidos con sumo sobre el tráfico, conseguir un resultado para la contaminación.

Obviamente tuve que consensuarlo con mi superior, ya que cómo he explicado previamente en este TFG no solo trabajaba para mí, también lo hacía para Hiberia.

Llegados a este punto nació un último problema. En la base de datos de Valencia se almacenaba un único registro de la contaminación a la una de la madrugada. Esto significa que no tenía con que comparar mis resultados puesto que no tiene nada que ver los niveles que se puedan alcanzar en hora punta, a los obtenidos a esas horas de la noche.

Es en este punto donde se encuentra el proyecto en estos momentos.



# **6 Conclusiones y trabajo futuro**

---

## **6.1 Conclusiones**

En este apartado procedo a comentar mis sensaciones finales tras la elaboración del TFG, tanto a nivel personal como técnico.

En lo que a nivel técnico se refiere me siento muy contento, ya que he sido capaz de aplicar muchos de los conceptos aprendidos durante la carrera. Si bien esto no ha sido suficiente, ya que en mi opinión, durante los años de los años de estudio nos enseñan una base y lo que es más importante, nos enseñan a buscarnos la vida ante las adversidades que surgen.

En mi caso llegaba con una serie de conocimientos respecto a lenguajes de programación, tecnologías...pero es que durante este periodo he tenido que formarme en muchas otras de las cuales no había oído hablar nunca. Es ahí donde uno es consciente de la formación que ha recibido y por lo cual estoy muy agradecido.

Además me he involucrado por primera vez en mi vida en un proyecto a nivel laboral dentro de una empresa, siendo totalmente responsable del desarrollo del mismo. Esto ha supuesto un reto personal, ya que el inicio me generó muchas dudas de que pudiese sacarlo adelante, pero poco a poco me iba adaptando a las circunstancias.

De esta manera he sido participe de un proyecto europeo, viendo cómo funciona una empresa, como se trabaja en equipo, como interactúan unas empresas con otras...por lo que considero que el desarrollo de este TFG ha sido una gran experiencia para mí y ha ayudado a mejorar mi formación.

## **6.2 Trabajo futuro**

Cómo he comentado en repetidas ocasiones a lo largo de este documento, no se ha alcanzado el desarrollo completo de este proyecto por los motivos expuestos. Es por ello que el primer objetivo no puede ser otro que terminarlo.

Una vez terminado lo correspondiente a este servicio, es decir, al cálculo de rutas menos contaminantes en una ciudad, el siguiente paso sería aplicarlo en alguna aplicación que nos pudiese aportar, por ejemplo:

- Predicciones de futuros días venideros, teniendo en cuenta parámetros que esta aplicación ya contempla cómo tipo de día, clima, tipo de vehículos en la ciudad...

- Ofrecer un servicio que te ofrezca la ruta más óptima en cuanto a relación contaminación-tiempo se refiere.

- Desarrollar la parte correspondiente al front-end, de manera que se pueda ofrecer una aplicación web al usuario donde permita interactuar con la aplicación, introduciendo filtros como la climatología, tipo de día, hora...



# Referencias

---

[1] Javier Gregori, Aumentan en España las muertes por contaminación atmosférica, Cadena ser, 11 de Octubre de 2017.

[http://cadenaser.com/ser/2017/10/11/ciencia/1507724875\\_838890.html](http://cadenaser.com/ser/2017/10/11/ciencia/1507724875_838890.html)

[2] Noemí Navas, Madrid estrena un programa pionero de medición de la contaminación de los vehículos en tiempo real, El país, 3 de Mayo de 2018.

[https://cincodias.elpais.com/cincodias/2018/05/03/companias/1525347662\\_156377.html](https://cincodias.elpais.com/cincodias/2018/05/03/companias/1525347662_156377.html)

[3] Joan Ferran, 9 apps sostenibles para cuidar del medio ambiente desde tu Smartphone, Blog HolaLuz 30 de Marzo de 2017

<https://blog.holaluz.com/9-apps-sostenibles-para-cuidar-del-medio-ambiente-desde-tu-smartphone/>

[4] OpenWeather.

<http://api.openweathermap.org/data/2.5/weather?id=524901&APPID=bfa3b12c20d8203bcbec99ef4767cbae&q=Madrid,ES>

[5] Ayuntamiento de Madrid.

<https://datos.madrid.es/portal/site/egob/>

[6] Elasticsearch.

<https://www.elastic.co/guide/index.html>

[7] MongoDB

<https://docs.mongodb.com/>

[8] Robomongo

<https://robomongo.org/>

[9] API Rest Google

<https://chrome.google.com/webstore/category/extensions?hl=es>

[10] Crontab

<https://help.ubuntu.com/community/CronHowto>

[11] Documentación SUMO.

[http://sumo.dlr.de/wiki/Simulation\\_of\\_Urban\\_MObility\\_-\\_Wiki](http://sumo.dlr.de/wiki/Simulation_of_Urban_MObility_-_Wiki)

[12] Google Maps.

<https://www.google.es/maps/@40.4837681,-3.7130504,15z>

---



## Glosario

---

API	Interfaz de programación de aplicaciones
REST	Transferencia de estado representacional
Base de datos	Colección de información organizada
Base relacional	Conjunto de tablas que contienen datos provistos en categorías predefinidas
Base no relacional	Conjunto de documentos sin estructura fija e indexados
JSON	JavaScript Object Notation
XML	Lenguaje de marcado extensible
Sistema Operativo	Conjunto de programas de un Sistema informático que gestiona los recursos del mismo
Algoritmo Dijkstra	Algoritmo del camino mínimo
Open data	Datos de las administraciones públicas accesibles
Datasets	Representación de datos residente en memoria
Multiparadigma	Que soporta más de un paradigma
Id	Identificador único
SmartPhone	Teléfono inteligente
Hardware	Conjunto de elementos físicos que componen un Sistema informático
Software	Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas
Memoria RAM	Dispositivo de una máquina donde se almacenan datos o instrucciones
Disco duro	Dispositivo de almacenamiento de datos

