



Universidad
Zaragoza

Trabajo Fin de Grado

Desarrollo e implementación de herramientas de monitorización y simulación de tráfico basadas en datos abiertos

Development of tools based on open data in order to monitorize and simulate traffic flows

Autor

David Sáez García

Directora

Raquel Trillo Lado

Escuela de Ingeniería y Arquitectura
2019

AGRADECIMIENTOS

En primer lugar quiero dar mi agradecimiento a Raquel Trillo por haberme brindado la oportunidad de realizar este proyecto bajo su supervisión. Además de ser una excelente directora, me ha ofrecido la oportunidad de colaborar en un proyecto real como TRAFAIR, interaccionando con sus investigadores y siguiendo las orientaciones recibidas. Tampoco quiero olvidarme de mandar mis agradecimientos a dichos investigadores, en especial de Sergio Ilarri.

En segundo lugar me gustaría agradecer a mi familia y mis amigos el apoyo moral que me han ofrecido en los meses que ha durado el proyecto, sin el cual la realización de dicho proyecto habría sido mucho más complicada.

En último lugar, me gustaría agradecer a todos los docentes de la Universidad de Zaragoza que a lo largo de estos años me han impartido clase o han tenido contacto conmigo, puesto que todos ellos me han aportado conocimientos muy valiosos.

Resumen

Desarrollo e implementación de herramientas de monitorización y simulación de tráfico basadas en datos abiertos

El objetivo del proyecto es proporcionar herramientas de predicción y simulación de tráfico, para entender cómo influyen los flujos de tráfico en la contaminación del aire. Inicialmente se querían desarrollar también herramientas de monitorización de tráfico, sin embargo no ha sido posible por motivos legales. Para cumplir con dicho objetivo, se ha desarrollado un sistema cuya arquitectura está definida en cuatro capas: el modelo de datos, el modelo de tráfico, el modelo de emisiones y una aplicación web.

En la base de datos se dispone de datos históricos (hasta marzo de 2019) sobre flujos de tráfico en Zaragoza. Además, también se dispone de un mapa de la ciudad de Zaragoza exportado de OpenStreetMap.

Para el modelo de tráfico se ha utilizado el simulador SUMO, que permite realizar simulaciones de tráfico de un día en concreto. Para su uso se han diseñado una serie de scripts en Python, encargados de recuperar la información de la base de datos, predecir el tráfico para el día indicado, adaptar esta información al formato requerido por SUMO y lanzar las simulaciones. La salida de este modelo es un fichero CSV que contiene el número de coches que ha pasado por cada calle de la ciudad en cada una de las 24 horas del día simulado.

Respecto al modelo de emisiones, en esta capa se ha utilizado el paquete VEIN, con el cual se calcula la cantidad de emisiones de NO_x producido por los flujos de tráfico calculados por SUMO. Para ello se ha creado un script en R que toma como entrada la salida de SUMO y que ofrece como salida las emisiones de NO_x en formato CSV, conteniendo el NO_x acumulado por cada calle de la ciudad en cada una de las 24 horas del día simulado.

Finalmente, se ha diseñado una aplicación web llamada TraFlow. El objetivo de la aplicación es que el usuario disponga de una interfaz amigable para realizar y visualizar los resultados de las simulaciones, sin necesidad de utilizar directamente los scripts de Python y R.

En conclusión, se han diseñado una serie de herramientas que permiten predecir y simular flujos de tráfico y las emisiones de NO_x asociadas a estos dentro de la ciudad de Zaragoza.

Índice

Lista de Figuras	VII
Lista de Tablas	IX
1. Introducción y objetivos	1
1.1. Introducción	1
1.2. Objetivos	2
1.3. Arquitectura del sistema	2
1.4. Herramientas y tecnologías utilizadas	4
1.5. Organización del documento	4
2. Modelo de tráfico	7
2.1. Datos disponibles	7
2.2. Estructura general del modelo de tráfico	9
2.3. Creación del fichero de red de carreteras	12
2.4. Creación del tráfico para SUMO	13
2.4.1. Predicción del tráfico	13
2.4.2. Cálculo de rutas para cada estación	14
2.4.3. Elección de rutas en base a los flujos reales	16
2.5. Validación	18
2.5.1. Error de estimación	18
2.5.2. Vehículos teletransportados	20
2.5.3. Velocidad media	22
2.6. Conclusiones	23
3. Modelo de emisiones	27
3.1. Cálculo de emisiones	27
3.2. Problemas abordados y decisiones tomadas	29
3.2.1. La flota de vehículos de Zaragoza	29
3.2.2. La temperatura ambiente	30

3.3. Validación	30
4. TraFlow	33
4.1. Arquitectura	33
4.2. El modelo de datos	34
4.3. La API REST	34
4.4. La interfaz de usuario	35
5. Gestión de riesgos y plan de seguridad del sistema	39
5.1. MAGERIT	39
5.1.1. Activos	39
5.1.2. Amenazas	42
5.1.3. Riesgos	44
5.1.4. Medidas de salvaguarda	45
5.2. Plan de mantenimiento	47
6. Conclusiones	49
6.1. Evaluación personal	49
6.2. Comparación del sistema desarrollado con los desarrollados en Módena y Santiago	50
6.3. Metodología y esfuerzos invertidos	50
6.4. Trabajo futuro	51
Bibliografía	53
Anexos	54
A. SUMO	57
B. Tipos de vía	59
C. Correspondencias de VEIN con la información de la base de datos	61
D. Mapa de navegación de TraFlow	63
E. Magerit detallado	71
F. Esquema Entidad-Relación de la base de datos de TRAFAIR	77

Lista de Figuras

1.1. Arquitectura general del sistema.	3
2.1. Realización de una simulación en SUMO.	10
2.2. Diagrama de secuencia de una simulación.	10
2.3. Creación del fichero de red.	12
2.4. Fichero de red visto desde la interfaz gráfica de SUMO. Se ve la plaza Basilio Paraíso además de otras calles.	13
2.5. Creación de los ficheros .cp.	15
2.6. Tasa de error total.	19
2.7. Error máximo.	19
2.8. Tasa de error por hora.	20
2.9. Tasa de coches teletransportados en total.	21
2.10. Teletransportes máximos.	21
2.11. Tasa de coches teletransportados por hora.	21
2.12. Velocidad media de los vehículos.	22
2.13. Velocidad media de los vehículos por hora.	22
2.14. Ejemplo de un atasco.	24
2.15. Total de teletransportes en una simulación. El naranja se corresponde a una red retocada.	24
2.16. Teletransportes por hora en una simulación. El naranja se corresponde a una red retocada.	25
3.1. Cálculo de emisiones.	28
4.1. Arquitectura de la aplicación web.	33
4.2. Diagrama de clases que representa la base de datos.	35
4.3. Documentación vista en Swagger-UI.	36
4.4. Página principal de la aplicación.	36
4.5. Mapa por colores que muestra los flujos.	37
5.1. Interrelación entre los activos.	41

5.2.	Degradación de los activos ante un ataque DDoS.	43
5.3.	Impacto a partir del valor y la degradación.	43
5.4.	Impacto de un ataque DDoS.	44
5.5.	Tabla de riesgos.	45
5.6.	Resumen de la amenaza de ataque de denegación de servicio.	46
6.1.	Diagrama de Gantt.	50
D.1.	Mapa de navegación de la aplicación.	63
D.2.	Ventana inicial de la aplicación.	64
D.3.	Calendario mostrado en la aplicación.	64
D.4.	Elección de hora.	64
D.5.	Aviso de que no hay datos.	65
D.6.	Aviso de que se ha lanzado simulación.	65
D.7.	Aviso de que se solo hay datos de flujos.	66
D.8.	Mapa mostrando flujos por colores.	67
D.9.	Opción de cambiar de mapa.	68
D.10.	Mapa mostrando acumulación de NOx por colores.	69
E.1.	Resumen de la amenaza de intrusión.	73
E.2.	Resumen de la amenaza de inundación.	74
E.3.	Resumen de la amenaza de avería.	75
F.1.	Tipo de entidad vehículo.	77
F.2.	Tipo de entidad factores de emisión.	77
F.3.	Tipo de entidad road.	78
F.4.	Tipo de entidad sensor.	78

Lista de Tablas

2.1.	Ejemplos de descripción de estaciones.	8
2.2.	Ejemplos de mediciones de las estaciones.	8
2.3.	Ejemplos de aristas de OSM.	8
2.4.	Ejemplos de nodos de OSM.	8
2.5.	Ejemplos de restricciones de giro de OSM.	8
2.6.	Ejemplo de csv de salida de SUMO.	11
2.7.	Ejemplo de datos utilizados para generar el modelo.	14
3.1.	Ejemplo de salida del script calculate_emissions.R.	28
3.2.	Ejemplos sobre la flota de Zaragoza.	29
3.3.	Temperaturas medias por mes en Zaragoza (año 2018).	31
5.1.	Activos del sistema.	40
5.2.	Escala de valores para valorar los activos respecto a una dimensionalidad	40
5.3.	Valoración de activos respecto a las dimensionalidades.	42
5.4.	Probabilidad de materialización de amenazas.	42
5.5.	Valores escalares para la probabilidad.	43
5.6.	Riesgo de la amenaza DDoS.	44
5.7.	Salvaguardas para un ataque DDoS.	45
6.1.	Horas invertidas en el proyecto.	51
B.1.	Peso para cada uno de los tipos de vía.	59
C.1.	Correspondencias de la clase de vehículo.	61
C.2.	Correspondencias del carburante.	61
C.3.	Correspondencias de la cilindrada.	62
C.4.	Correspondencias del estándar de emisión.	62
E.1.	Amenaza de inundación.	71
E.2.	Amenaza de fallo en BD.	71
E.3.	Amenaza de intrusión.	72

E.4. Amenaza de denegación de servicio.	72
---	----

Capítulo 1

Introducción y objetivos

En este capítulo se expone el contexto en el que se engloba el proyecto, además de dar una pequeña introducción de todo el trabajo que se ha realizado. Por otro lado, se exponen brevemente los objetivos que tiene el proyecto.

1.1. Introducción

Se estima que unas 400.000 muertes al año se deben a la contaminación en Europa, convirtiéndola en la primera causa de muerte ambiental en Europa. Entre las principales fuentes de contaminación ambiental se encuentra el tráfico vial, lo que ha provocado que se empiecen a tomar medidas en las ciudades, tanto nacionales como internacionales. Sin embargo, las administraciones públicas carecen de herramientas precisas para estimar el nivel de contaminación en función del flujo de tráfico.

Por otro lado, se da la circunstancia de que en febrero de 2017 la Comisión Europea advirtió a España de las violaciones continuas de la normativa de la contaminación del aire. Reiteradamente se incumple el límite de contaminación del aire para el dióxido de nitrógeno (NO₂), cuya mayoría de emisiones parten del tráfico rodado.

El proyecto TRAFAIR¹ surge para proporcionar herramientas precisas que permitan analizar la situación actual con el objetivo de tomar medidas que permitan mejorar la calidad del aire a nivel urbano.

El presente trabajo se desarrolla en el marco del proyecto TRAFAIR en Zaragoza y tiene como objetivo principal el diseño y desarrollo de herramientas para estimar los flujos de tráfico y la contaminación producida por estos dentro de la ciudad de Zaragoza.

¹<http://trafair.eu/>

1.2. Objetivos

El objetivo principal del trabajo es el desarrollo e implementación de herramientas de predicción y simulación de tráfico, para entender cómo afectan los diferentes flujos de tráfico a la contaminación del aire.

Para la consecución de dicho objetivo, el proyecto se dividió en varias etapas que se detallan a continuación:

- Estudio de metodologías y herramientas a utilizar en el proyecto. En esta fase se estudiaron además diferentes fuentes de datos con información relevante para el trabajo como por ejemplo el portal de datos abiertos de Zaragoza.
- Análisis de requisitos del sistema. En esta fase se han identificado qué necesidades tiene el sistema desarrollado y qué herramientas son más adecuadas para cubrirlas.
- Desarrollo del modelo de tráfico, estimando los flujos de tráfico, empleando como base la herramienta SUMO.
- Desarrollo del modelo de emisiones, que dados unos flujos de tráfico estima las emisiones que estos producen, empleando como base la herramienta VEIN.
- Desarrollo de la aplicación web TraFlow, para permitir al usuario final realizar simulaciones y visualizar sus resultados de forma sencilla.
- Análisis de gestión de riesgos del sistema desarrollado y diseño del plan de seguridad de este.

Inicialmente se querían desarrollar también herramientas de monitorización de tráfico, sin embargo, no ha sido posible ya que la Ley Orgánica de Protección de Datos no permite que le sean entregados al proyecto TRAFAIR datos en tiempo real sobre el estado del tráfico, debido a que se podría llegar a identificar las rutas de ciertos vehículos.

1.3. Arquitectura del sistema

En la figura 1.1 se detalla la arquitectura a alto nivel del sistema desarrollado. Se observa que hay 4 partes diferenciadas: el modelo de datos, el modelo de tráfico, el modelo de emisiones y TraFlow.

El modelo de tráfico se encarga de predecir y simular flujos de tráfico. Por otro lado, el modelo de emisiones se encarga de calcular las emisiones de NO_x (NO + NO₂)

producidas por los flujos de tráfico. Finalmente, TraFlow es una aplicación web que permite al usuario final realizar y visualizar simulaciones de forma sencilla.

En la figura se observa que modelo de tráfico toma como entrada la información de la base de datos. Además, también se observa el modelo de emisiones toma como entrada la salida del modelo de tráfico y, que TraFlow hace uso de ambos modelos.

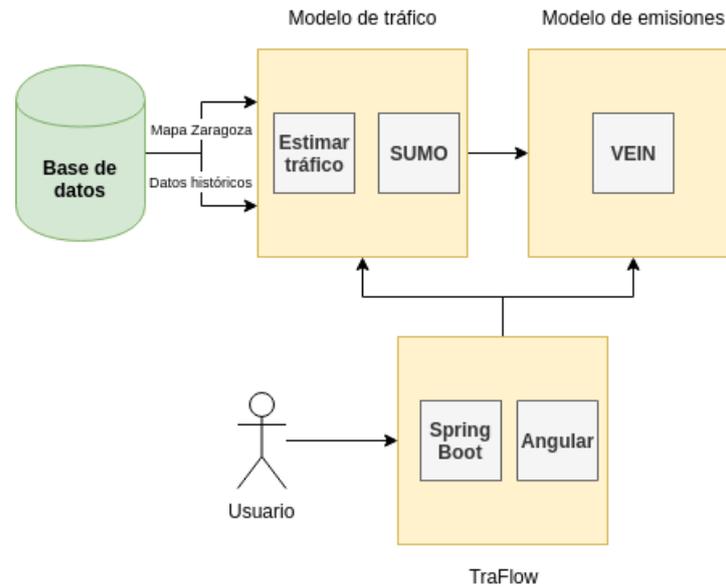


Figura 1.1: Arquitectura general del sistema.

1.4. Herramientas y tecnologías utilizadas

A continuación, se detallan las diferentes herramientas que se han empleado a lo largo del desarrollo de este trabajo, incidiendo en aquellas que han resultado más relevantes:

- OpenStreetMap (OSM, [1]): Proyecto colaborativo para crear mapas editables y libres. Concretamente, se utiliza el mapa de la ciudad de Zaragoza como base para realizar las simulaciones y predicciones de flujos de tráfico.
- Simulation of Urban Mobility (SUMO, [2]): Paquete de simulación de tráfico que permite simular cómo un conjunto de coches se mueve dentro de una red de carreteras.
- Vehicular Emissions INventories (VEIN, [3]): Paquete de R [4] que permite estimar las emisiones producidas por los distintos tipos de vehículos.
- PostgreSQL: Sistema Gestor de Bases de Datos Relacional. En el marco del proyecto TRAFAIR se ha diseñado un modelo de datos que en el caso de Zaragoza se ha implementado con este sistema.
- PostGIS: Complemento de PostgreSQL que facilita la gestión de datos espaciales.
- Spring Boot: Tecnología basada en Java empleada en el desarrollo de aplicaciones web.
- Bootstrap: Framework CSS y JavaScript utilizado para el diseño de interfaces.
- Angular: Framework para crear y mantener aplicaciones web de una sola página.

Por otro lado, se han utilizado 4 lenguajes de programación distintos: Python, R, Java y TypeScript. Además, se han utilizado los entornos PyCharm, IntelliJ, WebStorm y RStudio.

1.5. Organización del documento

El trabajo realizado se expone en 5 capítulos, además de un último capítulo para las conclusiones del proyecto. En el capítulo 2 se explica el modelo de tráfico que se ha implementado para poder simular y predecir flujos de tráfico. En el capítulo 3 se explica el modelo de emisiones implementado, el cual se encarga de calcular las emisiones de NOx generadas por los flujos de tráfico obtenidos en el modelo de tráfico. En el capítulo 4 se detalla la aplicación web que se ha creado para poder realizar simulaciones

y predicciones y visualizar resultados de forma sencilla e intuitiva. En el capítulo 5 se explica el análisis de riesgos y el plan de seguridad del sistema realizado. Finalmente, en el capítulo 6 se exponen las conclusiones sobre el proyecto.

Por otro lado, hay una serie de anexos complementarios a este documento en los que se encuentra información adicional sobre varios aspectos del trabajo realizado.

Finalmente, cabe destacar que todo el código desarrollado durante el proyecto se encuentra en el repositorio de Gitlab <https://gitlab.com/trafair/trafair-private>.

Capítulo 2

Modelo de tráfico

En este capítulo se describe el modelo de tráfico que se utiliza en el proyecto, detallándose los pasos que se han seguido para la generación de dicho modelo y las decisiones que han sido tomadas.

2.1. Datos disponibles

En la ciudad de Zaragoza existen 46 estaciones permanentes. Una estación permanente es un dispositivo que se encuentra activo las 24 horas del día en un determinado punto de una vía, y que se encarga de medir el número de coches que pasan por dicha vía en un determinado periodo de tiempo configurable. En el caso de Zaragoza dicho periodo es de una hora. La ubicación de estas estaciones es fija y no cambia a lo largo del tiempo. El proyecto TRAFAIR dispone en su base de datos (ver anexo F) de los datos recopilados por estas estaciones desde el 01-01-2018 hasta el 24-03-2019. Como ya se ha comentado en la introducción, no es posible acceder a datos en tiempo real por motivos legales.

Por otro lado, se dispone del mapa de Zaragoza exportado desde OSM y almacenado en la base de datos de TRAFAIR. Dicho mapa se utilizará para modelar las calles de Zaragoza dentro de SUMO. No solo se almacenan las calles, sino que también se almacena otros datos como la ubicación de semáforos, el número de carriles de una calle o la velocidad máxima dentro de la calle. También se almacena la información relativa a las restricciones de giro, es decir, a qué calles se puede avanzar desde una calle determinada, direcciones que no pueden tomarse, etc.

En las tablas 2.1, 2.2, 2.3, 2.4 y 2.5 se muestran algunos ejemplos de los datos de los que se dispone y que se encuentran almacenados en la base de datos de TRAFAIR para la realización del presente trabajo.

id	direction	geom	road
EP6.2	t	0101...4440	92741399
EP19.1	t	0101...4440	116606357

Tabla 2.1: Ejemplos de descripción de estaciones.

id_sensor_traffic	datetime	flow
EP6.2	2018-03-10 00:59:59	130
EP19.1	2018-06-21 04:59:59	92

Tabla 2.2: Ejemplos de mediciones de las estaciones.

id	speed_limit	one_way	num_lanes	type	geom
4759680	50	t	3	primary	0...4
3129443	30	t	1	residential	0...4

Tabla 2.3: Ejemplos de aristas de OSM.

id	geom	traffic_light
30389381	0...A	t
30389382	0...A	f

Tabla 2.4: Ejemplos de nodos de OSM.

id	type	id_road_node_via	id_road_section_from	id_road_node_to	id_road_section_via
27872	only_right_turn	288554225	26337732	31721827	NULL
545214	only_straight_on	685350410	54516732	397870776	NULL

Tabla 2.5: Ejemplos de restricciones de giro de OSM.

2.2. Estructura general del modelo de tráfico

El fin del modelo de tráfico es obtener por cada segmento de una vía el número de coches, junto a su velocidad media, que han pasado en cada hora del día a simular. Esto se hace a partir de los datos que se tienen a disposición, puesto que no resulta viable instalar estaciones de medición en todos los segmentos de carretera. Por ejemplo, si se simula un día completo, por cada segmento habrá 24 valores (uno por cada hora) que indican el número de coches que han pasado por ese segmento y su velocidad media.

Para obtener dichos valores se utiliza el simulador SUMO. Este simulador necesita de dos ficheros para realizar una simulación: un fichero de red (extensión *.net.xml*) que modele la ciudad sobre la que se desea realizar la simulación y un fichero que defina los vehículos junto a las rutas que siguen (extensión *.rou.xml*). En el anexo A se explica en mayor detalle el funcionamiento del simulador y la estructura de dichos ficheros. Además, se han creado varios scripts escritos en Python para facilitar el uso del simulador desde la línea de comandos sin necesidad de interactuar con la interfaz gráfica proporcionada por SUMO.

Las figuras 2.1 y 2.2 ilustran cómo realizar una simulación. El script *do_simulation.py* toma como entrada un fichero de red (fichero *.net.xml*) y una serie de rutas (ficheros *.cp*) que pueden seguir los vehículos, además del día para el que se quieren estimar los flujos. Una ruta consiste en una lista ordenada de segmentos, en la cual cada segmento es alcanzable dentro del fichero de red por su predecesor en la lista. Además, es posible definir una hora entre 0 y 23, para que así se hagan las estimaciones de tráfico solo para esa hora en vez de para las 24 horas del día que se ha indicado.

Cabe destacar que en caso de simular un día completo, se lanzan 24 simulaciones independientes de una hora y se agregan los resultados. Se ha optado por esta alternativa y no por la de lanzar una sola simulación de 24 horas para reducir la complejidad de la simulación.

Por cada simulación que se realice, en primer lugar se realizará una predicción, en base a datos históricos, del tráfico esperado para el día que se ha indicado. Dicha demanda de tráfico se almacenará en un fichero XML (fichero *.rou.xml*) para que pueda ser interpretada por el simulador.

Una vez realizada una simulación, como salida ha de obtenerse un fichero que indique el número de coches que han pasado (y su velocidad media) por cada uno de los segmentos de la red en cada hora del día simulado. En los siguientes apartados se explica la creación del fichero de red, la creación de los ficheros de rutas y la creación de los ficheros que definen los vehículos.

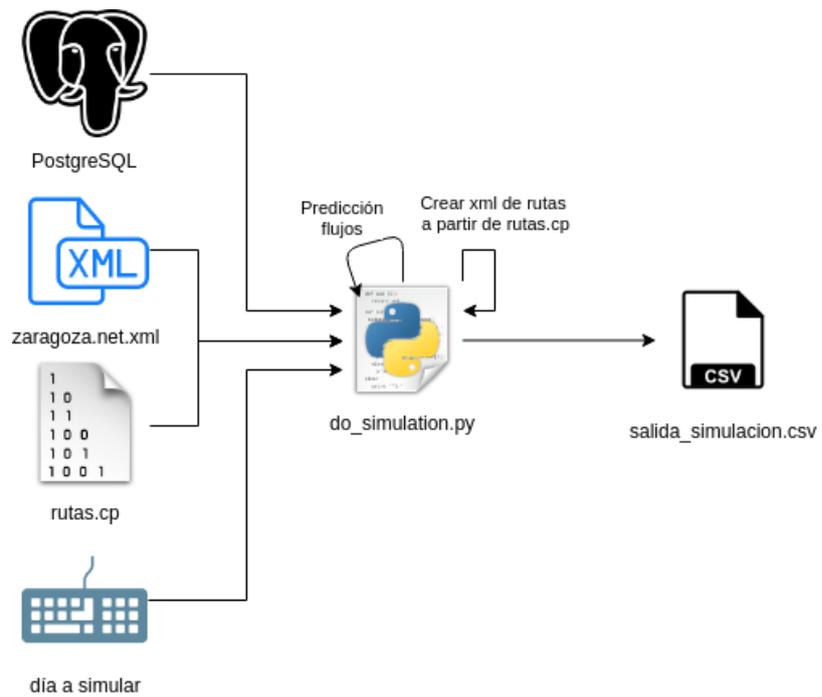


Figura 2.1: Realización de una simulación en SUMO.

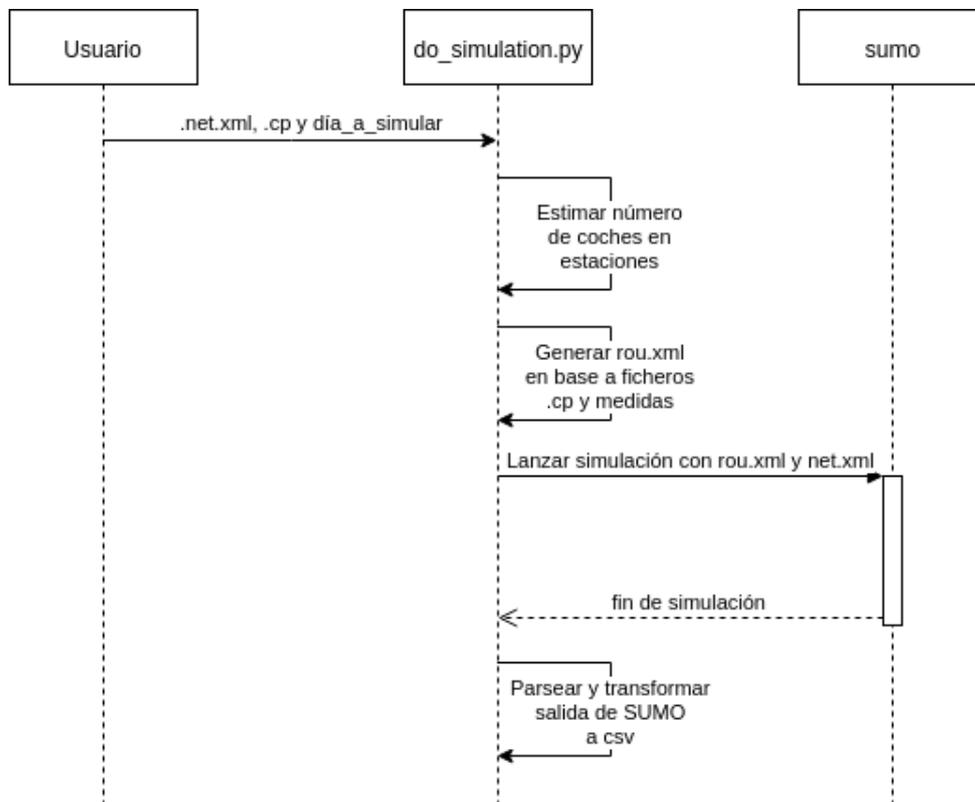


Figura 2.2: Diagrama de secuencia de una simulación.

edge	hour	flow	mean_speed (km/h)
85773358	15	354	85.7
85773358	16	310	83.1
92741248	15	248	40.1
92741248	16	230	42.8

Tabla 2.6: Ejemplo de csv de salida de SUMO.

2.3. Creación del fichero de red de carreteras

La figura 2.3 ilustra cómo se genera el fichero de red de carreteras:

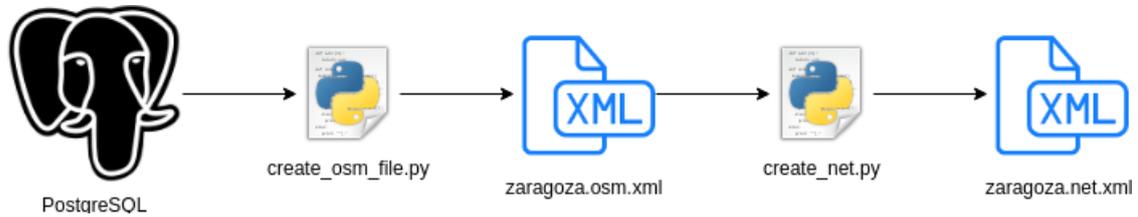


Figura 2.3: Creación del fichero de red.

El fichero *create_osm_file.py* es un script escrito en Python que se encarga de obtener la información relativa al mapa de Zaragoza de la base de datos de TRAFAIR. A partir de dicha información se genera un fichero (extensión *.osm.xml*) con la red de carreteras en el formato especificado por OSM. En la figura 2.4 se muestra un fichero de red de carreteras visualizado desde la interfaz gráfica de SUMO.

Una vez obtenido el fichero OSM, se proporciona este como entrada al script *create_net.py* que, haciendo uso de la herramienta *netconvert* incluida con el software de SUMO, transforma el fichero OSM a una red de carreteras compatible con SUMO.

La herramienta *netconvert* permite generar ficheros de red a partir de otros formatos, como *shapefiles*. Sin embargo, se ha decidido utilizar como entrada ficheros de OSM por varios motivos:

- OSM es un sistema colaborativo y su uso es libre y gratuito.
- En la base de datos de TRAFAIR se dispone de información de la ciudad de Zaragoza exportada desde OpenStreetMap.
- OSM almacena, entre otras cosas, varios datos sobre las calles. Se utilizan datos sobre las coordenadas geográficas, la velocidad máxima permitida, la ubicación de semáforos, etc.
- Al transformar un fichero en cualquier formato al formato requerido por SUMO, se crean nuevos identificadores para los segmentos de las calles e intersecciones. Al realizar dicha transformación desde un fichero de OSM, es posible almacenar las referencias a los identificadores originales. De esta forma pueden obtenerse los flujos de tráfico para cada uno de los segmentos de las calles, asociadas a un identificador determinado, que están almacenadas en la base de datos.

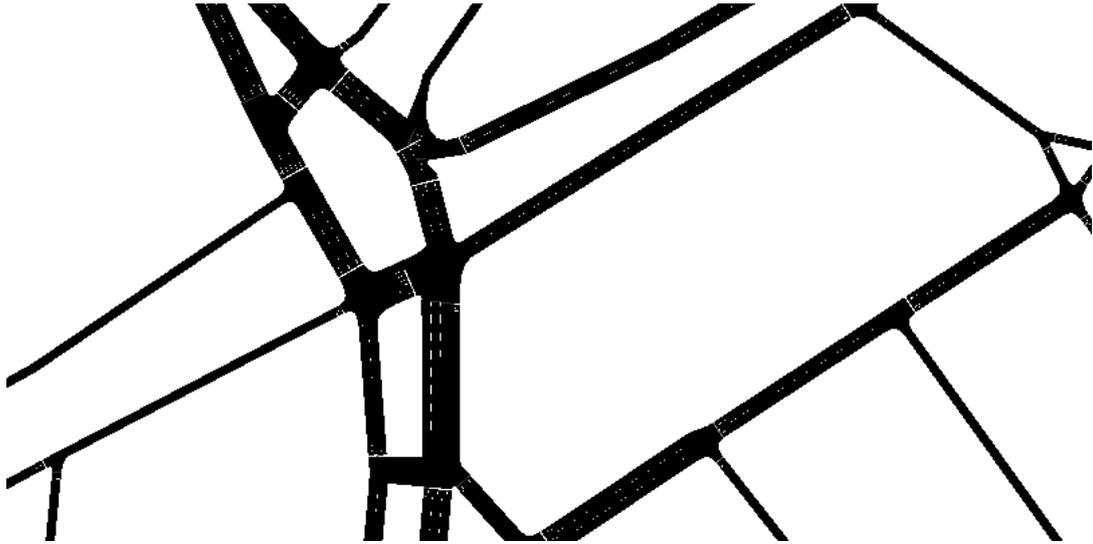


Figura 2.4: Fichero de red visto desde la interfaz gráfica de SUMO. Se ve la plaza Basilio Paraíso además de otras calles.

2.4. Creación del tráfico para SUMO

Se entiende por tráfico a un conjunto de vehículos que se mueven dentro de la red. Cada uno de estos vehículos sigue una ruta finita, que no es más que una sucesión de aristas (o segmentos), siendo una arista (o segmento) una calle (o parte de una calle) dentro del simulador. Dichas rutas no pueden tener aristas repetidas. Además, estos vehículos tienen definido un instante de salida, que es un entero mayor que 0, que indica el instante (paso de la simulación) en el que el coche entrará en la simulación. Por ejemplo, si se realiza una simulación de una hora y el valor de dicho entero para un vehículo es de 1800 el vehículo iniciará su ruta en el paso 1800 de simulación.

Dicho tráfico se crea cada vez que se lanza una simulación, y varía en función del día a simular. La solución implementada se divide en tres partes: predicción del tráfico, cálculo de rutas para cada estación y elección de rutas según los flujos reales.

Cabe destacar que inicialmente las dos últimas partes no iban a implementarse, ya que SUMO ofrece una serie de herramientas que permiten generar el tráfico deseado de forma sencilla. Sin embargo, se detectaron y experimentaron varias ineficiencias con estas herramientas, por lo que se descartó su uso. En la sección *Conclusiones* se detallan estas ineficiencias.

2.4.1. Predicción del tráfico

Como ya se ha comentado en secciones anteriores, solo se dispone de datos de flujos de tráfico medidos por las estaciones desde el 01-01-2018 hasta el 24-03-2019. Además, dentro de este intervalo de tiempo hay algunos días en los que alguna estación no

ha recogido datos. Si bien es cierto que está previsto que para el proyecto se sigan entregando periódicamente datos de nuevos días, es interesante e imprescindible que la herramienta sea capaz de predecir flujos de tráfico para días para los cuales no se tienen datos.

Por tanto, se ha generado un modelo que permite estimar los flujos de tráfico para días desconocidos. Este modelo está basado en una regresión lineal múltiple. El valor de R^2 -ajustado que ha arrojado el modelo ha sido de 0.7736, es decir, más del 75 % de la varianza es explicada por el modelo, lo que se ha considerado suficiente como para utilizarlo.

Los datos utilizados para generar el modelo han sido las propias medidas de las estaciones permanentes, disponibles en la base de datos de TRAFAIR, e información sobre la fecha en la que se realizó la medida, concretamente el mes en el que se realizó, la hora en la que se realizó y el tipo de día en el que se realizó, siendo el tipo de día o “entre semana”, o “sábado” o “festivo” (incluye a los domingos). Para la obtención de los días festivos se ha consultado el calendario laboral de la ciudad de Zaragoza. En la tabla 2.7 se incluye un ejemplo de estos datos.

Por tanto, estimar el tráfico que pasará por una estación concreta un día concreto a partir del modelo de regresión generado es trivial, simplemente hay que pasarle a la función de predicción el nombre de la estación, el mes del día a estimar, la hora que se quiere estimar y el tipo de día, y ofrecerá como salida el flujo estimado.

station	month	hour	type_day	flow
EP2.1	enero	0	E	300
EP2.1	marzo	8	S	581
EP2.1	marzo	20	F	344

Tabla 2.7: Ejemplo de datos utilizados para generar el modelo.

2.4.2. Cálculo de rutas para cada estación

El objetivo es generar una serie de rutas que puedan ser empleadas y utilizadas por los vehículos dentro del simulador. Para ello, inicialmente hay que averiguar la arista del mapa (fichero *.net.xml*) donde se encuentra cada estación temporal (a partir de ahora la denominaremos arista objetivo). En la tabla 2.1 el campo *road* indica la calle de OSM en la que se encuentran las estaciones. A partir de dicho identificador, es trivial obtener el identificador de la arista equivalente dentro de la red de carreteras de SUMO (fichero *.net.xml*) ya que, recordemos, dicho fichero es generado a partir de un fichero OSM.

Posteriormente, por cada una de las aristas objetivo (46 en total) se generan todas las rutas posibles que pasan por ellas. Estas rutas han de tener una longitud máxima (en número de aristas) que tiene que ser indicada por el usuario. Por ejemplo, si suponemos que el usuario ha indicado que quiere rutas de longitud máxima 30, se generan todas las rutas de longitud 30 o menos que pasan por la primera arista objetivo, seguidamente se generan todas las que pasan por la segunda de las aristas objetivo, etc. Además, para cada una de estas rutas se almacena el tiempo (en segundos) que se tarda en alcanzar la arista objetivo desde la arista inicial de la ruta. Dicho tiempo se calcula en base a dos heurísticas:

- Los vehículos circulan siempre a la velocidad máxima que permite la arista en la que se encuentran.
- Los semáforos se encuentran siempre en verde.

Con estas dos heurísticas, se tiene que el tiempo (en segundos) que se tarda en recorrer una arista es igual a su longitud (en metros) dividido entre la velocidad máxima permitida en dicha arista (en metros/segundo).

Como las rutas son siempre las mismas independientemente del día a simular, se ha decidido crear un script llamado *create_paths.py* que cree y almacene las rutas en ficheros (un fichero por cada estación) con extensión *.cp*, para que dichos ficheros sean leídos cuando se realiza una simulación. De esta forma, se evita tener que realizar el cálculo de las rutas cada vez que se realiza una simulación. En la figura 2.5 se ilustra el funcionamiento de

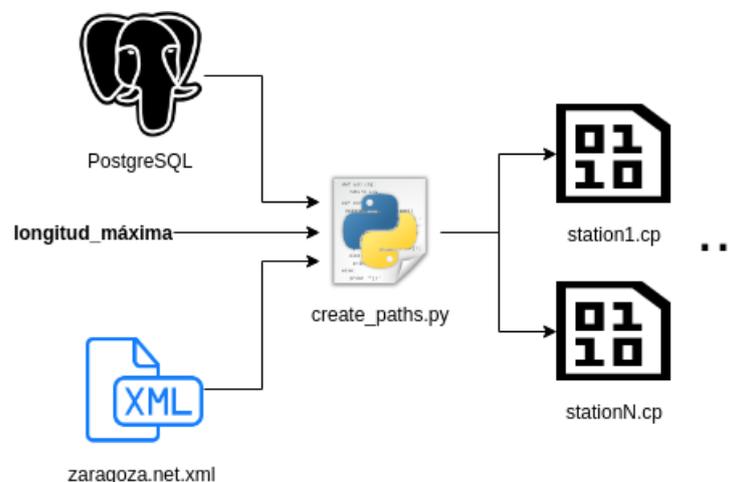


Figura 2.5: Creación de los ficheros .cp.

2.4.3. Elección de rutas en base a los flujos reales

Una vez calculadas y almacenadas las rutas para las 46 estaciones, ya es posible generar el tráfico para las simulaciones. Por ejemplo, si se quiere simular una única hora de un día determinado, y se tiene que en ese día a esa hora en la estación *EP2.1* van a pasar 100 vehículos, entonces hay que generar 100 vehículos y asignarles aleatoriamente 100 de las rutas calculadas previamente para dicha estación, y así sucesivamente para el resto de estaciones.

- En cuanto a la elección de caminos, a estos se les asigna un peso, en función de su longitud (en número de aristas) y del tipo de aristas que contienen. De esta forma se consigue que no se escojan caminos cortos (por ejemplo con una única arista) y que aquellos caminos que contienen determinados tipos de vías, como avenidas o autovías, tengan más probabilidad de ser escogidos. El tipo de vía es extraído de OSM (ver tabla 2.3). En el anexo B se encuentra una tabla con el peso por cada tipo de vía.
- Por otro lado, el paso de los vehículos por la arista objetivo se reparte uniformemente a lo largo del tiempo. En el ejemplo anterior (100 coches en una hora) el reparto se haría de forma que pase un coche por la arista objetivo cada 36 segundos. Como con las rutas también se almacena el tiempo estimado que se tarda en alcanzar la arista objetivo desde la arista inicial de la ruta, a partir del instante en el que el vehículo entra en la simulación puede estimarse en qué momento el vehículo alcanzará la arista objetivo.
- Finalmente, cabe destacar que los vehículos generados son siempre coches de pasajeros. SUMO permite definir además otro tipo de vehículos, como autobuses o camiones. Sin embargo, aunque se tiene información sobre la flota de vehículos de Zaragoza se ha decidido generar solo vehículos de pasajeros. En el modelo de emisiones sí se tendrá en cuenta la flota de vehículos.

Una vez generados los vehículos con sus respectivas rutas, se almacena dicha información en un fichero *.rou.xml* para poder cargar el tráfico en dentro del simulador.

Listing 2.1: Ejemplo de fichero *.rou.xml* con un único coche definido

```
<routes>
  <vType id="type1" accel="0.8" decel="4.5" sigma="0.5"
    length="5" maxSpeed="70" />

  <vehicle id="0" type="type1" depart="0">
    <route edges="beg_middle_end_rend" />
  </vehicle>
</routes>
```

```
</vehicle>
</routes>
```

Ajuste de los flujos

El método explicado en las secciones anteriores para generar tráfico no es totalmente preciso por dos motivos:

- El tráfico se genera de forma secuencial por cada una de las estaciones, independientemente de las demás. Esto implica, por ejemplo, que el tráfico generado para la estación *A* también puede pasar por la estación *B*, lo que provocaría un exceso de tráfico en la estación *B*. Lo mismo ocurre a la inversa.
- Las heurísticas empleadas. No siempre les será posible a los vehículos circular a la velocidad máxima de la vía ni se encontrarán todos los semáforos en verde, lo que implica que el tiempo estimado en una ruta para llegar desde la arista inicial hasta la arista objetivo será por lo general menor que el tiempo real. De este modo, habrá vehículos que nunca lleguen a pasar por la arista objetivo ya que la simulación finalizará antes de que lleguen a dicha arista.

Para abordar este problema, se utilizan unos dispositivos de SUMO llamados *calibradores*, que se definen en ficheros XML y que se pasan como parámetro a SUMO. Estos dispositivos se encargan de regular el tráfico por hora en la arista en la que se hayan definido. Por ejemplo, si se define un calibrador en la arista *190281#1* y se especifica que el flujo ha de ser de 500 vehículos por hora, el calibrador insertará o eliminará coches de la arista cuando considere que no va a alcanzarse o va a superarse dicho flujo. El algoritmo utilizado para decidir cuándo insertar o eliminar vehículos se describe en [5].

Las rutas a seguir por los vehículos insertados por los calibradores pueden ser o fijas o aleatorias, según se quiera configurar. En este trabajo se ha optado por utilizar rutas aleatorias.

Listing 2.2: Ejemplo de fichero XML con un calibrador

```
<additional>
  <vType id="t0" speedDev="0.1" />
  <routeProbe id="cali_edge1_probe" edge="edge1" freq="60"
    file="output.xml" />
  <calibrator id="cali_edge1" lane="edge1_0" pos="0"
    output="detector.xml" freq="60"
    routeProbe="cali_edge1_probe">
    <route id="cali1_fallback" edges="edge1" />
    <flow begin="0" end="3600" route="cali_fallback">
```

```
    vehsPerHour=" 2500" speed=" 27.8" type=" t0"
    departPos=" free" departSpeed=" max" />
</calibrator>
</additional>
```

2.5. Validación

Para validar el modelo se han empleado tres métricas que se definen a continuación:

- *Error de estimación.* El error de estimación en una estación a una hora determinada es la diferencia (en valor absoluto) entre el tráfico esperado y el tráfico que realmente ha pasado en la simulación. Por ejemplo, si se tiene que en la estación *EP2.1* tienen que pasar 500 coches en una hora y en la simulación han pasado 420, tenemos un error de 80. Se medirá el error en cada una de las 46 estaciones por cada hora simulada.
- *Vehículos teletransportados.* Un vehículo es teletransportado a la siguiente arista de su camino por el simulador SUMO cuando lleva tanto tiempo parado que el simulador determina que se debe a un problema de la simulación y no a una situación normal de tráfico. Ese tiempo es por defecto de 300 segundos, aunque es configurable.
- *Velocidad media* a la que circulan los vehículos en la simulación. Esta se utilizará para comparar el rendimiento entre los dos modelos.

Por otro lado, SUMO permite utilizar dos modelos de tráfico distintos para realizar simulaciones: un modelo mesoscópico [6] y un modelo microscópico [7] (ver anexo A para ver diferencias entre ambos modelos). Por ello, se comparará una simulación mesoscópica de un día con una microscópica del mismo día para determinar cuál es más eficaz considerando las 3 métricas anteriormente descritas.

2.5.1. Error de estimación

La *tasa de error total* por día simulado es la suma total de errores de todas las estaciones en cada una de las 24 horas de simulación dividido entre la demanda de vehículos total esperada. Como se observa en la figura 2.6, el error es menor en la simulación mesoscópica. Es decir, en la simulación mesoscópica la demanda generada difiere un 0,6% de la demanda esperada, mientras que en la simulación microscópica difiere un 1,8%.

El *error máximo* se define como el mayor error sucedido en una estación a una hora determinada un día concreto. En la figura 2.7 se observa que el error máximo es mayor en la simulación microscópica.

Por otro lado, en la figura 2.8 se ha obtenido la tasa de error por hora, comparándose con el número de coches que hay en el simulador en cada hora. Se observa que la tasa de error aumenta considerablemente cuando el número de coches es mayor en el simulador, siendo esta diferencia aún más notable en la simulación microscópica.

En definitiva, los datos muestran que el error es considerablemente menor en la simulación mesoscópica. Además, en escenarios en los que la demanda de tráfico es menor el error en la simulación mesoscópica mejora aún más respecto del de la simulación microscópica.



Figura 2.6: Tasa de error total.



Figura 2.7: Error máximo.



Figura 2.8: Tasa de error por hora.

2.5.2. Vehículos teletransportados

La *tasa de coches teletransportados* es la suma total de coches que han sido teletransportados en la simulación, dividido por la demanda total de vehículos esperada. Como se observa en la figura 2.9, en la simulación mesoscópica no ha habido ningún coche teletransportado, mientras que el 1% de los coches han sido teletransportados en la simulación microscópica.

El número mayor de teletransportes se obtiene de escoger el número máximo de teletransportes ocurridos en una arista en una hora concreta. En la figura 2.10 se observa que este valor es de 44 para la simulación microscópica.

Por otro lado, en la figura 2.11 se ha obtenido la tasa de coches teletransportados por hora, comparándose con el número de coches que hay en el simulador en cada hora. En el caso de la simulación microscópica se observa que la tasa aumenta cuando la demanda es mayor. En la sección 2.6 se explica el principal motivo por el que se producen estos teletransportes en el modelo microscópico.

En definitiva, respecto a esta métrica la simulación mesoscópica es considerablemente mejor que la simulación microscópica, pues en la primera no se producen teletransportes de coches, mientras que en la segunda sí, siendo estos más frecuentes cuando la demanda de tráfico aumenta.

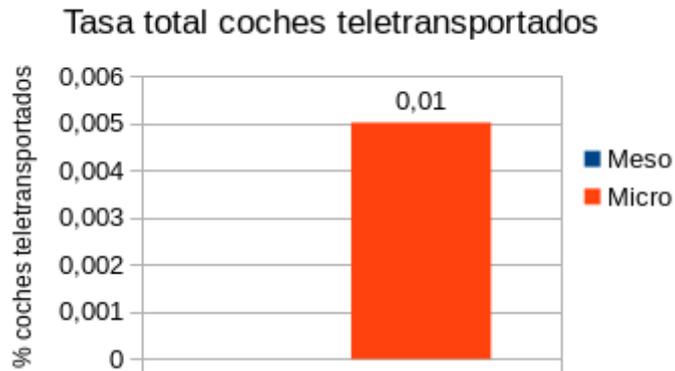


Figura 2.9: Tasa de coches teletransportados en total.



Figura 2.10: Teletransportes máximos.



Figura 2.11: Tasa de coches teletransportados por hora.

2.5.3. Velocidad media

Finalmente, se ha utilizado la velocidad media para comparar ambos modelos. En las figuras 2.12 y 2.13 se observa que la velocidad media de los vehículos es considerablemente mayor en el modelo mesoscópico. Esto se debe a los teletransportes comentados en el apartado anterior, ya que para que un coche sea teletransportado previamente ha tenido que estar un tiempo parado 300 segundos sin poder moverse, lo que implica que la velocidad media de los vehículos disminuya.

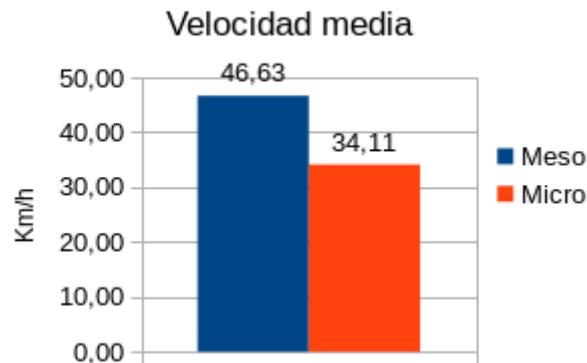


Figura 2.12: Velocidad media de los vehículos.

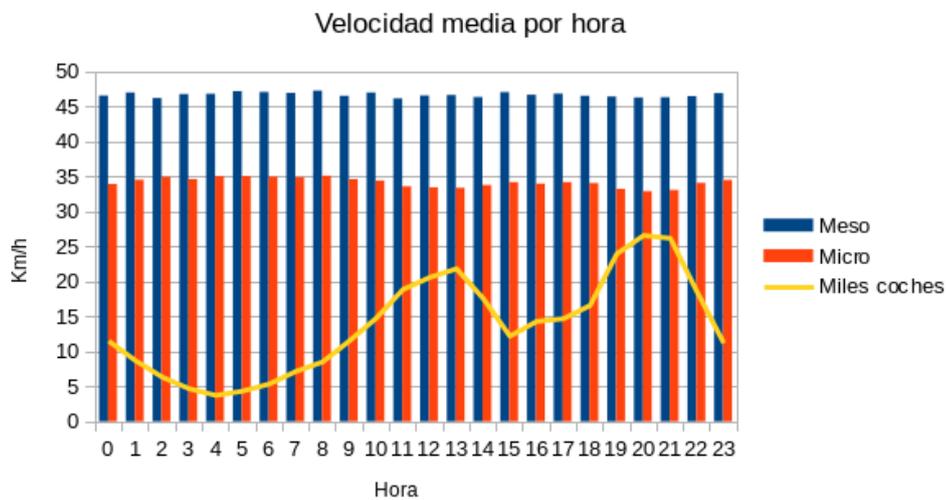


Figura 2.13: Velocidad media de los vehículos por hora.

A la vista de los datos y las gráficas de los 3 apartados anteriores, queda claro que el modelo mesoscópico es más eficaz a la hora de simular tráfico que el modelo microscópico, por lo que se ha determinado que es el modelo que se utilizará en las simulaciones.

2.6. Conclusiones

En este apartado van a enumerarse y explicarse brevemente los principales problemas que se han encontrado durante el desarrollo del modelo de tráfico:

- Falta de información en OSM. La parte negativa de que OSM esté mantenido por la comunidad es que hay información incompleta. Por ejemplo, faltan semáforos reales que no están en OSM, calles que no tienen indicado el número de carriles que poseen, etc. Se espera que esto se corrija en un futuro, ya que hay miembros del proyecto TRAFAIR que se encargan de añadir información de la ciudad de Zaragoza a OSM.
- Ineficiencia de los calibradores. Inicialmente se consideró utilizar únicamente calibradores (explicados en secciones anteriores) para generar los flujos de tráfico. Sin embargo, se experimentó que no eran capaces de generar la demanda requerida por sí solos, por lo que se optó por implementar el algoritmo de generación de rutas.
- La simulación microscópica. Inicialmente se configuró SUMO para utilizar un modelo microscópico de tráfico. En este modelo se modelan explícitamente los vehículos: estos van moviéndose por la red por sí solos, avanzando a otras calles cuando pueden, adelantando a otros vehículos, parando en los semáforos... Sin embargo, al lanzar una simulación microscópica se ha experimentado que se producen bloqueos, ya que en ocasiones los vehículos se quedan parados sin motivo aparente al principio las intersecciones. En la figura 2.14 se ilustra un ejemplo de un bloqueo. Este es debido a que el coche rojo quiere pasar al carril donde se encuentra el coche verde y a que el coche verde quiere pasar al carril donde se encuentra el coche rojo. Ambos vehículos detectan que hay otro vehículo en la posición a la que quieren cambiarse, por lo que se quedan esperando a que el otro vehículo se mueva, provocando así un atasco. Estos bloqueos no se liberan hasta que alguno de los dos coches es teletransportado por el simulador. Retocando la red de carreteras (fichero *.net.xml*) con la herramienta *netedit* puede solucionarse de forma manual este problema y otros similares. Se llegaron a retocar más de 70 intersecciones, pero finalmente se desechó esta alternativa por

la imposibilidad de automatizar el proceso de retocar la red, lo que hace que no sea un método escalable. En las figuras 2.15 y 2.16 se hace una comparativa entre los coches teletransportados utilizando una red sin retocar y una red mejorada.

Otros problemas menores han sido la conversión del fichero de OSM al formato de SUMO y el parseo y conversión de los ficheros de salida de SUMO a CSV.

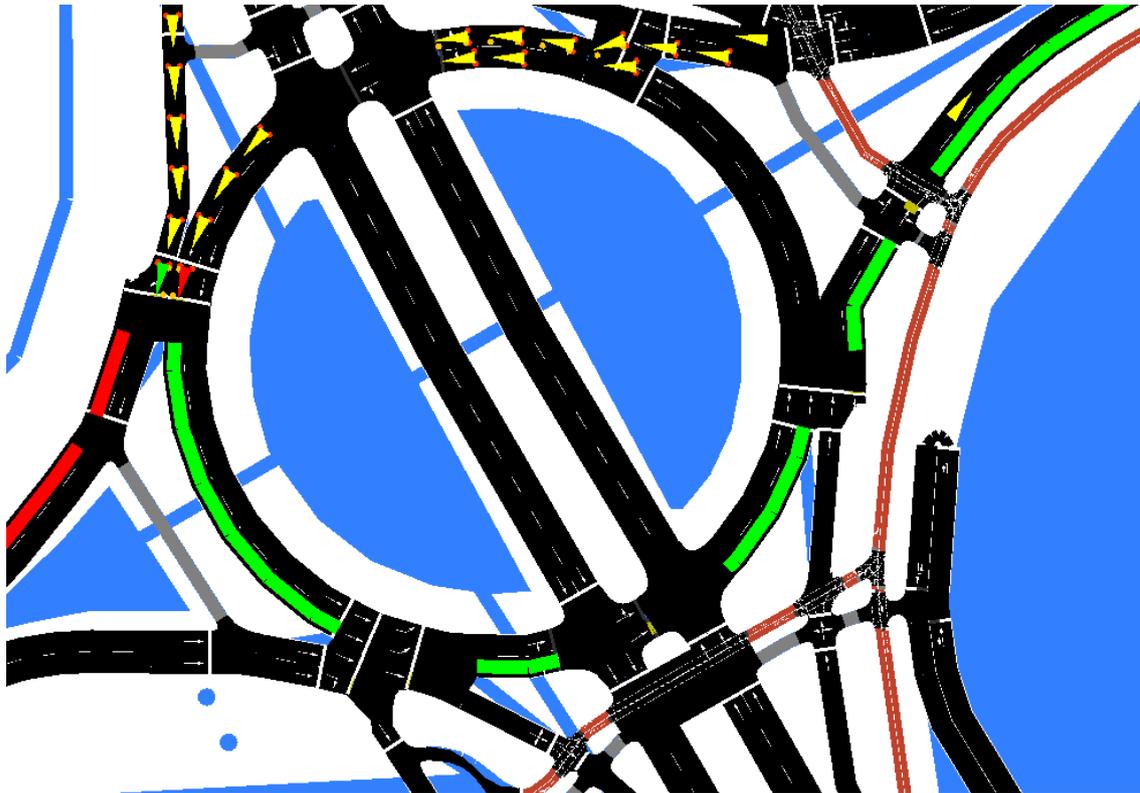


Figura 2.14: Ejemplo de un atasco.

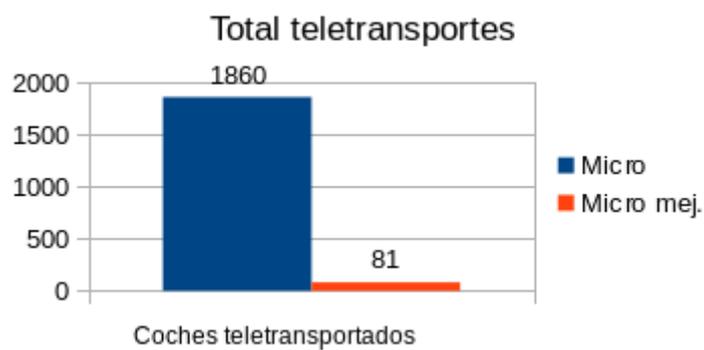


Figura 2.15: Total de teletransportes en una simulación. El naranja se corresponde a una red retocada.

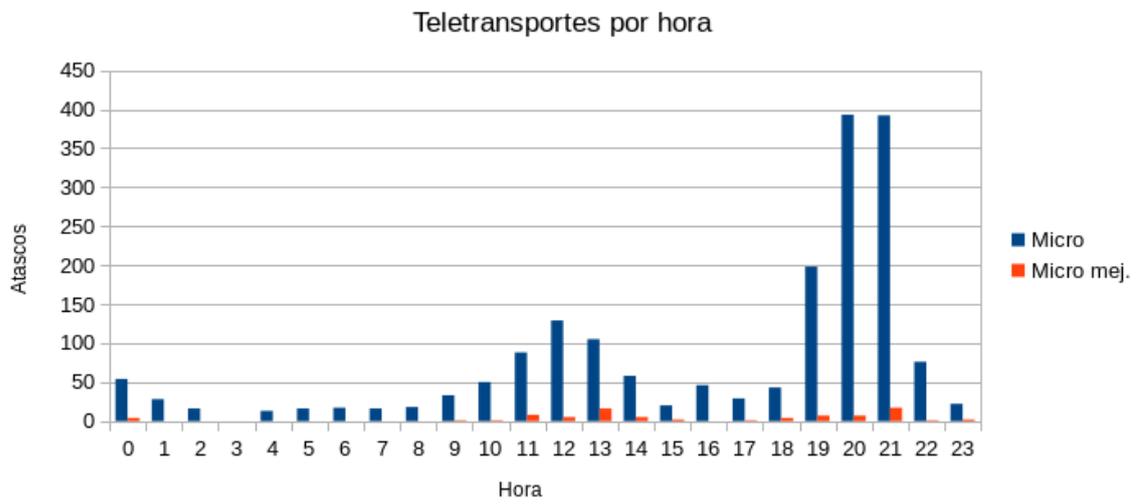


Figura 2.16: Teletransportes por hora en una simulación. El naranja se corresponde a una red retocada.

Capítulo 3

Modelo de emisiones

En este capítulo se describe el modelo de emisiones de tráfico que se utiliza en el proyecto, detallándose cómo se ha construido dicho modelo y qué decisiones han sido tomadas.

3.1. Cálculo de emisiones

El objetivo del modelo de emisiones es calcular la cantidad de NO_x, óxido nítrico y dióxido de nitrógeno, por distancia recorrida ($\frac{kg}{km}$) producido por los flujos de tráfico en cada una de las calles de la ciudad. Se consideran dos tipos de emisiones: emisiones en caliente y emisiones en frío. Las primeras se producen cuando el motor del vehículo se encuentra en su temperatura normal operativa, mientras que las segundas se producen en la fase de arranque del motor.

Para lograr dicho objetivo se ha utilizado VEIN [3], que es un paquete de R que permite estimar las emisiones producidas por un vehículo en función del tipo de vehículo, la velocidad a la que circula y la temperatura ambiente. Se ha optado por utilizar este paquete ya que es el que tiene mayor uso en el ámbito del medio ambiente. La figura 3.1 ilustra el flujo de cómo se calculan las emisiones:

- *sim_results.csv* es un fichero que contiene los resultados de una simulación realizada por SUMO.
- *zaragoza_fleet.csv* es un fichero que contiene información sobre la flota de vehículos de la ciudad de Zaragoza. Tiene definidos los diferentes tipos de vehículos que hay en la ciudad y el porcentaje que representan sobre el total de vehículos de la ciudad.
- *calculate_emissions.R* es un script desarrollado en R que, haciendo uso de VEIN, calcula las emisiones de NO_x por arista y hora.

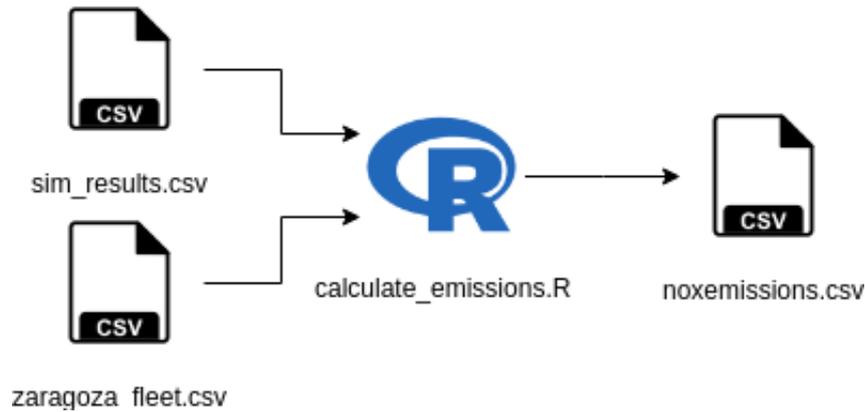


Figura 3.1: Cálculo de emisiones.

– *noxemissions.csv* es la salida del script R. La tabla 3.1 ilustra un pequeño ejemplo.

En definitiva, el script *calculate_emissions.R* lee el fichero *sim_results.csv* (ver tabla 2.6), que almacena el flujo de tráfico por arista en cada hora simulada, y teniendo en cuenta la flota de vehículos de Zaragoza genera como salida otro fichero csv que contiene el NOx que se ha generado por cada arista en cada hora. El pseudocódigo del algoritmo utilizado para el cálculo de emisiones se encuentra al final de la sección.

edge	hour	NOx [kg/km]
12981211	15	0.005
12981211	16	0.012
72124519	15	0.00391
72124519	16	0.00765

Tabla 3.1: Ejemplo de salida del script *calculate_emissions.R*.

Listing 3.1: Pseudocódigo del cálculo de emisiones

```

def emUnCoche(tipoCoche, velocidad, temperatura):
    # Funcion que calcula las emisiones, en frio y caliente,
    # producidas por un vehiculo que tiene un tipo
    # determinado y que circula a una velocidad concreta.
    # Hace uso de la libreria VEIN.
    return VEIN(tipoCoche, velocidad, temperatura)

def calcularEmisionArista(velocidad, flujo, temperatura):
    emisionTotal = 0
    for tipoCoche in zaragoza_fleet.csv:
        emisionUnCoche = emUnCoche(tipoCoche, velocidad,
                                    temperatura)
        # Porcentaje de coches del tipo "tipoCoche" que
        # hay en Zaragoza sobre el total de coches
        porcentaje = extraerPorcentaje(tipoCoche)
  
```

```

    emisionTipoCoche = emisionUnCoche * porcentaje
                        * flujo
    emisionTotal = emisionTotal + emisionTipoCoche
return emisionTotal

```

```

def main(temperaturaAmbiente):
    for entrada in sim_results.csv:
        arista = extraerArista(entrada)
        hora = extraerHora(entrada)
        velocidad = extraerVelocidad(entrada)
        flujo = extraerFlujo(entrada)
        emisionTotalArista = calcularEmision(velocidad, flujo,
                                             temperatura)
        escribirEnSalida(arista, hora, emisionTotalArista)

```

3.2. Problemas abordados y decisiones tomadas

En esta sección se van a exponer los principales problemas encontrados en el modelo de emisiones y qué decisiones se han tomado para solucionarlos.

3.2.1. La flota de vehículos de Zaragoza

A los miembros del proyecto TRAFair se les ha entregado información sobre la flota de vehículos de la ciudad de Zaragoza. En la tabla 3.2 se ven algunos de los datos que han sido proporcionados:

id	engine_size	fuel	emission_standard	class	number
412	1.6	diesel	euro 3	passenger_car	756
97	NULL	diesel	euro 2	heavy_duty_truck	115
270	0.5	petrol	euro 1	motorcycle	59

Tabla 3.2: Ejemplos sobre la flota de Zaragoza.

Es decir, la flota de vehículos de Zaragoza se divide en varios tipos de vehículos, que se diferencian por 4 características: el tamaño del motor, el tipo de carburante que utilizan, el estándar de emisión europeo y la clase. Además, el campo *number* indica el número de coches de ese tipo que hay registrados en Zaragoza.

Sin embargo, han surgido problemas respecto a estos datos:

- Los valores de las 4 características no coinciden con los que VEIN maneja. Afortunadamente, ha sido posible realizar una correspondencia con los valores almacenados en la base de datos y los valores que utiliza VEIN. En el anexo C se especifican estas correspondencias.

- Falta de información. Además de la categoría (campo *class* en la tabla 3.2) es necesario especificar la subcategoría del vehículo. Como no se dispone de esta información, la elección de la subcategoría para cada tipo de vehículo se ha realizado de forma aleatoria, eligiendo esta entre las subcategorías válidas para la categoría a la que pertenece el vehículo. Además, para los vehículos pesados (camiones y autobuses) también es necesario especificar el peso, por lo que se ha escogido el peso de forma similar a como se ha escogido la subcategoría.
- Tipos de vehículos inválidos. Tras haber realizado las correspondencias, seleccionado la subcategoría y especificado el peso (en el caso de los vehículos pesados), a la hora de utilizar VEIN han aparecido errores respecto a algunos tipos de vehículos. Concretamente, VEIN detecta que algunos tipos no son válidos y finaliza con error. La decisión que se ha tomado en estos casos (50 tipos de vehículos en total) ha sido la de descartar esos tipos supuestamente inválidos y, para los tipos válidos, ajustar el porcentaje de vehículos sobre el total que suponen.

3.2.2. La temperatura ambiente

En las secciones anteriores se ha mencionado que VEIN necesita el tipo de vehículo, la velocidad y la temperatura ambiente para calcular las emisiones producidas por el tipo de vehículo especificado. El tipo de vehículo y la velocidad ya se ha especificado cómo se obtienen, sin embargo aún no se ha comentado nada sobre la temperatura.

La solución empleada ha sido utilizar las temperaturas medias mensuales. Por ejemplo, si se van a calcular las emisiones para el 24 de marzo, utilizar la temperatura media del mes de marzo. Son dos los motivos por los que se ha decidido adoptar esta solución. El primero es que es una solución sencilla, y el segundo es que en la propia documentación de VEIN se recomienda adoptar esta solución. En la tabla 3.3 se especifican las temperaturas que se han utilizado.

3.3. Validación

Para validar los datos de polución de NO_x obtenidos se utilizarán datos reales de contaminación recogidos por sensores. Varios miembros de TRAFair se han encargado de desplegar una red de sensores por la ciudad de Zaragoza encargados de recoger datos de contaminación. Sin embargo, no ha sido posible validar los datos estimados por VEIN con datos reales pues en el momento en el que se ha realizado este trabajo la red de sensores no se encontraba operativa. Una vez se despliegue la red, otros miembros de

mes	temperatura media (°C)
enero	8.7
febrero	6.8
marzo	10.7
abril	14.7
mayo	18.4
junio	22.9
julio	27.1
agosto	26.4
septiembre	23.7
octubre	16.5
noviembre	11.6
diciembre	8.6

Tabla 3.3: Temperaturas medias por mes en Zaragoza (año 2018).

TRAFair se encargarán de validar los datos estimados por VEIN.

Capítulo 4

TraFlow

Una vez creados los scripts que permiten el cálculo de flujos de tráfico y las emisiones de NOx producidas por estos, se ha creado TraFlow, una aplicación web encargada por un lado de facilitar a los usuarios ambos cálculos, y por otro de mostrar los resultados en un mapa de la ciudad de Zaragoza. Así pues, el objetivo es que los usuarios puedan consultar el flujo de tráfico y el NOx producido por estos para el día que deseen, y en caso de que no se tenga esa información en la base de datos que sean capaces de lanzar una simulación para consultar los resultados una vez esta haya finalizado.

4.1. Arquitectura

La figura 4.1 refleja la arquitectura de la aplicación web:

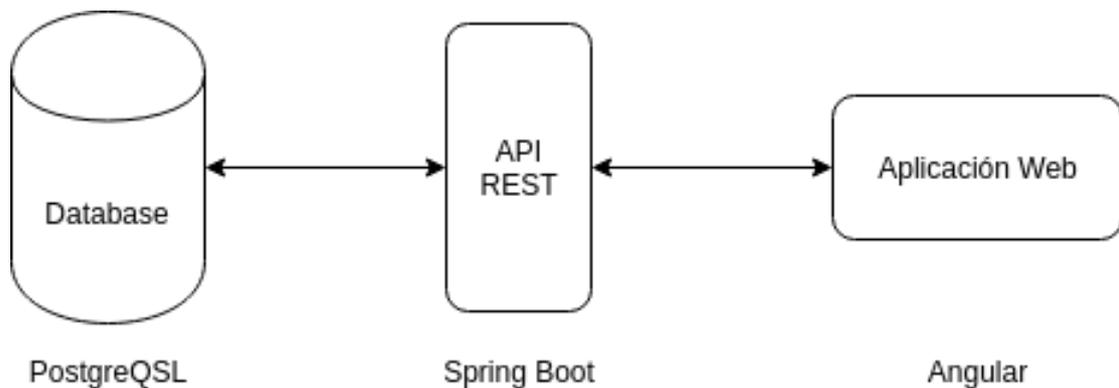


Figura 4.1: Arquitectura de la aplicación web.

Como puede observarse, se ha empleado el patrón MVC (Modelo-Vista-Controlador) con el objetivo de separar la interfaz de usuario del modelo de datos. De esta forma, cuando la aplicación desea un determinado dato se comunica con el controlador, y este se encarga de recuperar la información de la base de datos y transformarla a un formato que la aplicación sea capaz de entender. Además esta arquitectura permite que sea posible sustituir componentes por otros

equivalentes sin afectar a los demás componentes. Por ejemplo, si en un futuro se decide utilizar otra tecnología como Node.js con Express en vez de Spring Boot, dicho cambio no afectará ni a la base de datos ni a la aplicación web.

Respecto a las tecnologías utilizadas, se ha utilizado PostgreSQL como gestor de base de datos, Spring Boot para desplegar una API REST y Angular (además de Bootstrap) para el desarrollo de la interfaz de usuario.

4.2. El modelo de datos

Se han considerado cuatro tipos de entidad distintos:

- Tipo de entidad *Flow*. Representa la cantidad de coches (y su velocidad media) que han pasado por un determinado segmento de la ciudad durante una hora concreta.
- Tipo de entidad *FlowJob*. Representa una petición de un usuario de calcular los flujos de tráfico para un día en concreto.
- Tipo de entidad *Pollution*. Representa la cantidad de NOx producida por el paso de vehículos por un determinado segmento de la ciudad durante una hora concreta.
- Tipo de entidad *PollutionJob*. Representa una petición de un usuario de calcular el NOx producido por los flujos de tráfico para un día en concreto.

En la figura 4.2 se ilustra un diagrama de clases con las entidades implicadas:

4.3. La API REST

La API REST creada permite cuatro operaciones distintas:

- Recuperar los flujos de tráfico de un día a una determinada hora.
- Pedir el cálculo de los flujos de tráfico para un día en concreto. Para ello, hace uso de los scripts mencionados en el apartado de modelo de tráfico.
- Pedir el NOx producido por los flujos de tráfico de un día a una determinada hora. Para ello, hace uso de los scripts mencionados en el apartado de modelo de emisiones.
- Pedir el cálculo del NOx producido por los flujos de tráfico para un día en concreto.

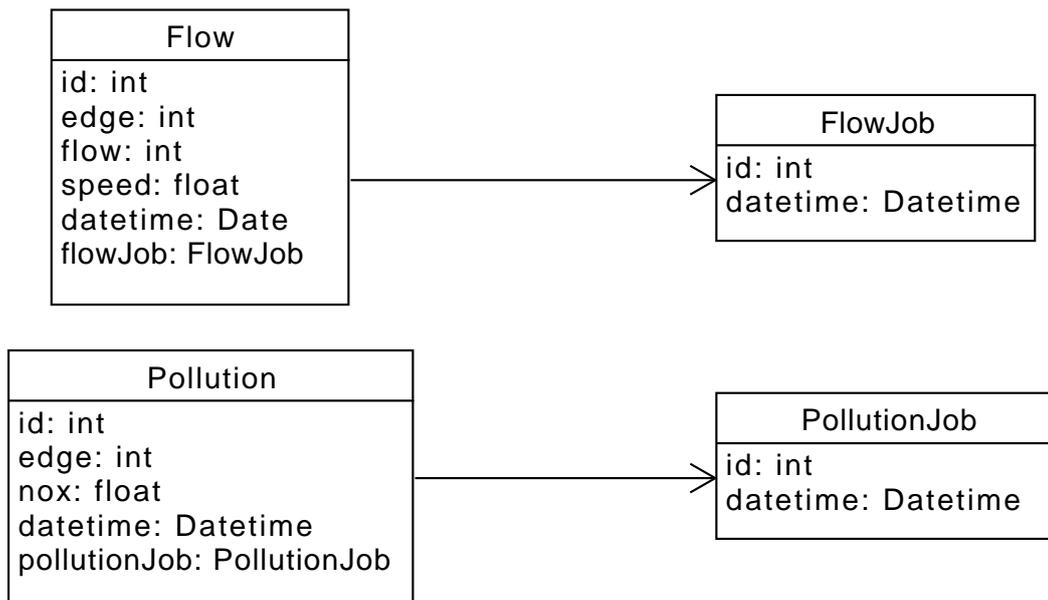


Figura 4.2: Diagrama de clases que representa la base de datos.

Además, la API se ha documentado con OpenAPI 3.0, con el objetivo de facilitar el trabajo a aquellas personas del proyecto TRAFAIR que en un futuro deseen mejorar la aplicación web. En la figura 4.3 puede verse parte de dicha documentación.

4.4. La interfaz de usuario

Finalmente, se ha desarrollado la interfaz de usuario mediante Angular y Bootstrap. La aplicación consiste básicamente en dos vistas: la página principal que permite escoger el día y la hora para el que se desea obtener el flujo de tráfico y el NOx producido, y la otra vista consiste en un mapa de la ciudad de Zaragoza en el que cada segmento se encuentra pintado de un color determinado, en función del tráfico o del nivel de NOx. A continuación se muestran dos imágenes de ejemplo. En el Anexo D se incluye un mapa de navegación de la aplicación.

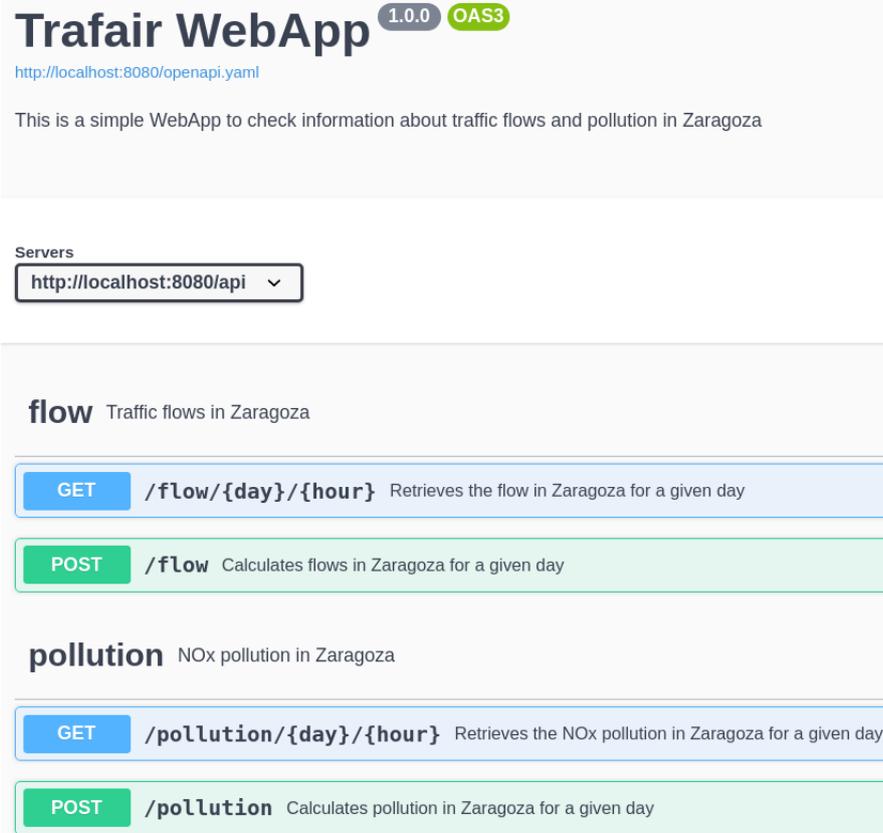


Figura 4.3: Documentación vista en Swagger-UI.

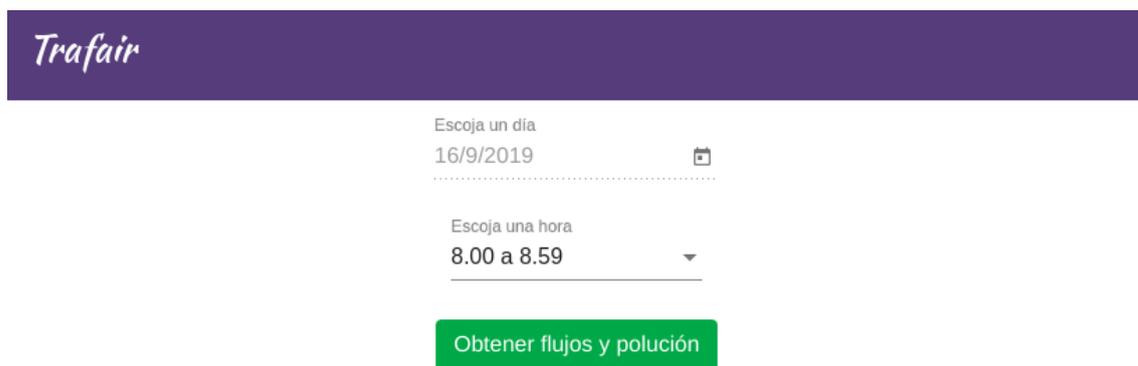


Figura 4.4: Página principal de la aplicación.



Figura 4.5: Mapa por colores que muestra los flujos.

Capítulo 5

Gestión de riesgos y plan de seguridad del sistema

Para llevar a cabo el análisis de riesgos y el plan de seguridad se ha utilizado la Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT, [8]), ya que es la que se utiliza dentro de la Administración Pública y además también es ampliamente utilizada en el ámbito de la empresa privada. Por otro lado, se detallará el plan de mantenimiento del sistema.

5.1. MAGERIT

Esta metodología tiene como objetivo concienciar de la existencia de riesgos y de la necesidad de gestionarlos, planificando el tratamiento oportuno para mantener los riesgos bajo control. Concretamente, se valorarán los activos respecto de las dimensiones disponibilidad, confidencialidad e integridad. En las siguientes secciones se exponen los activos identificados, las amenazas sobre dichos activos, los riesgos y las medidas de salvaguardas a implementar.

5.1.1. Activos

En la tabla 5.1 se identifican los activos del sistema. Se han clasificado según los grupos indicados por la metodología MAGERIT: esenciales (información, datos y servicios) y subordinados (activos necesarios para ofrecer la información, datos y servicios).

En la figura 5.1 se indica la interrelación de los distintos activos. Los tipos de activos se han representado mediante colores distintos, y se ha considerado que los daños en activos de la parte inferior de la figura repercuten en los activos de la parte superior a los que están conectados.

Por otro lado, se han valorado los activos según las dimensiones disponibilidad,

confidencialidad e integridad. Para ello, se ha considerado la escala cualitativa de valores de la tabla 5.2.

En la figura 5.3 se ilustra la valoración de los activos con respecto a las dimensiones disponibilidad, confidencialidad e integridad. Un valor de *Muy alto* implica que la dimensión tiene un impacto grande en el activo, mientras que un valor de *Muy bajo* implica que el impacto es mínimo.

Designación de activo	Tipo de activo
Servicio de visualización de simulaciones	Activo esencial (servicio)
Servicio de creación de simulaciones	Activo esencial (servicio)
Base de datos	Activo esencial (datos)
Software: aplicación Web	Activo subordinado (infraestructura)
Software: SUMO	Activo subordinado (infraestructura)
Software: VEIN	Activo subordinado (infraestructura)
Software: Sistema Gestor de Bases de Datos	Activo subordinado (infraestructura)
Hardware: servidor para aplicación web	Activo subordinado (infraestructura)
Hardware: servidor para base de datos	Activo subordinado (infraestructura)
Sala de servidores	Activo subordinado (instalaciones)
Administrador de la BD	Personal
Administrador del servidor web	Personal

Tabla 5.1: Activos del sistema.

Valor
Muy alto
Alto
Medio
Bajo
Muy bajo
No aplica

Tabla 5.2: Escala de valores para valorar los activos respecto a una dimensionalidad

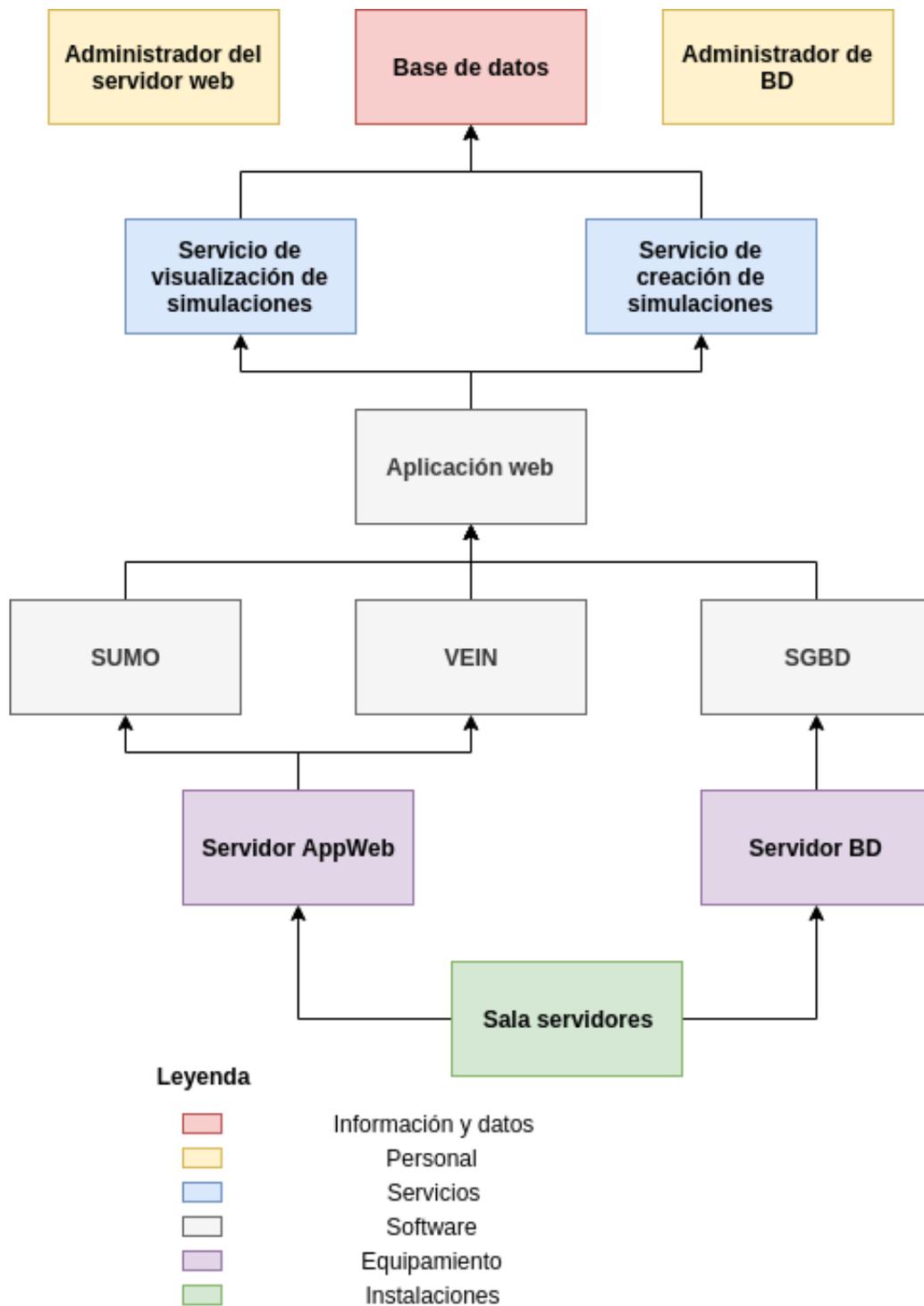


Figura 5.1: Interrelación entre los activos.

	Valor		
	Dimensión		
	D	C	I
Servicio de visualización de simulaciones	Muy alto	Bajo	Bajo
Servicio de creación de simulaciones	Muy alto	Bajo	Bajo
Base de datos	Muy alto	Medio	Muy alto
Software: aplicación Web	Alto	Medio	Bajo
Software: SUMO	Alto	Bajo	Bajo
Software: VEIN	Alto	Bajo	Bajo
Software: SGBD	Alto	Bajo	Bajo
Hardware: servidor para AppWeb	Alto	Bajo	Bajo
Hardware: servidor para BD	Alto	Bajo	Bajo
Sala de servidores	Bajo	Bajo	Bajo

Tabla 5.3: Valoración de activos respecto a las dimensionalidades.

5.1.2. Amenazas

Se han considerado 4 amenazas distintas:

- Inundación en la sala de servidores
- Avería de origen físico/lógico en el servidor donde se encuentra instalada la base de datos
- Intrusión de persona no autorizada en la base de datos
- Ataque de denegación de servicio

La descripción en detalle de estas amenazas se encuentra en el anexo E.

Una vez identificadas las amenazas, se detalla la probabilidad de que estas se materialicen (ver tabla 5.4). El valor de dichas probabilidades se basa en la escala cualitativa detallada en la tabla 5.5.

Amenaza	Probabilidad
Inundación	Muy infrecuente
Avería en BD	Normal
Intrusión	Infrecuente
DDoS	Normal

Tabla 5.4: Probabilidad de materialización de amenazas.

Una vez detalladas las amenazas y su probabilidad de materialización, el siguiente paso en la metodología es estimar la degradación sobre los activos que supondría la materialización de dichas amenazas, tomando dicha degradación tres posibles valores: 1 %, 10 % o 100 %. En la figura 5.2 se detalla la degradación que supondría sobre los

Probabilidad	Valor
1/100	Muy infrecuente
1/10	Infrecuente
1	Normal
10	Frecuente
100	Muy frecuente

Tabla 5.5: Valores escalares para la probabilidad.

activos un ataque de denegación de servicio. La degradación del resto de amenazas se encuentra en el anexo E.

Activo	Degradación dimensión		
	D	I	C
Servicio de visualización de simulaciones	_100%	_1%	_1%
Servicio de creación de simulaciones	_100%	_1%	_1%
Base de datos	_1%	_1%	_1%
Software: Aplicación Web	_100%	_1%	_1%
Software: SUMO	_1%	_1%	_1%
Software: VEIN	_1%	_1%	_1%
Software: SGBD	_1%	_1%	_1%
Hardware: servidor AppWeb	_1%	_1%	_1%
Hardware: servidor BD	_1%	_1%	_1%
Sala de servidores	_1%	_1%	_1%

Figura 5.2: Degradación de los activos ante un ataque DDoS.

Finalmente, se estima el impacto de materialización de una amenaza sobre los activos. Dicho valor se calcula a partir del valor el activo y la degradación que produce la amenaza sobre el activo, según los valores que aparecen en la figura 5.3. El impacto estimado para la amenaza de ataque de denegación de servicio se encuentra en la figura 5.4. El impacto del resto de amenazas sobre los activos se encuentra en el anexo E.

valor	degradación			
	_1%	_10%	_100%	na
Muy alto	M	A	MA	na
Alto	B	M	A	na
Medio	MB	B	M	na
Bajo	MB	MB	B	na
Muy bajo	MB	MB	MB	na
na	na	na	na	na

Figura 5.3: Impacto a partir del valor y la degradación.

Activo	Impacto dimensión		
	D	I	C
Servicio de visualización de simulaciones	MA	M	M
Servicio de creación de simulaciones	MA	M	M
Base de datos	M	M	M
Software: Aplicación Web	A	B	B
Software: SUMO	B	B	B
Software: VEIN	B	B	B
Software: SGBD	B	B	B
Hardware: servidor AppWeb	B	B	B
Hardware: servidor BD	B	B	B
Sala de servidores	MB	MB	MB

Figura 5.4: Impacto de un ataque DDoS.

5.1.3. Riesgos

Una vez estimadas las amenazas, el siguiente paso en la metodología es estimar el riesgo. En MAGERIT el riesgo se define como el impacto sobre los activos por la probabilidad de ocurrencia de las amenaza. A partir de esos dos valores y la tabla de la figura 5.5 se calcula el riesgo. La tabla 5.6 muestra el riesgo que supone la materialización de un ataque de denegación de servicio sobre los activos. El riesgo del resto de amenazas se encuentra en el anexo E.

Activo	Riesgo		
	D	C	I
Servicio de visualización de simulaciones	MA	M	M
Servicio de creación de simulaciones	MA	M	M
Base de datos	M	M	M
Software: Aplicación Web	A	B	B
Software: SUMO	B	B	B
Software: VEIN	B	B	B
Software: SGBD	B	B	B
Hardware: servidor AppWeb	B	B	B
Hardware: servidor BD	B	B	B
Sala de servidores	MB	MB	MB

Tabla 5.6: Riesgo de la amenaza DDoS.

	probabilidad					
	1/100	1/10	1	10	100	na
impacto	Muy infrecuente	Infrecuente	Normal	Frecuente	Muy frecuente	na
MA	A	MA	MA	MA	MA	na
A	M	A	A	MA	MA	na
M	B	M	M	A	A	na
B	MB	B	B	M	M	na
MB	MB	MB	MB	B	B	na
na	na	na	na	na	na	na

Figura 5.5: Tabla de riesgos.

5.1.4. Medidas de salvaguarda

Tras el cálculo del riesgo, el siguiente paso en la metodología consiste en estimar medidas de salvaguarda para cada una de las amenazas. Estas salvaguardas tienen como objetivo proteger el sistema. En la tabla 5.7 se detallan las medidas de salvaguarda consideradas para la amenaza del ataque DDoS.

Una vez realizado el primer análisis de riesgos es necesario determinar la eficacia de las medidas de salvaguarda planteadas respecto a cada uno de las dimensiones, para minimizar la probabilidad de materialización de las potenciales amenazas o reducir su impacto en caso de que se materialicen. Para evaluar la eficacia de las medidas de salvaguarda se utilizará una escala cualitativa (moderan el riesgo, minimizan el riesgo, eliminan el riesgo, no corrigen y no aplica). La eficacia de las salvaguardas planteadas agregadas se encuentra en la figura 5.6. Finalmente se estima el riesgo residual, el cual también detalla en la figura 5.6.

Salvaguarda 1	Migración de los servidores a un proveedor de servicios cloud IaaS, como AWS
Salvaguarda 2	Contratar un sistema de protección anti DDoS
Justificación	Los ataque de denegación de servicio son relativamente frecuentes. Un DDoS con éxito provocaría que los usuarios no pudieran entrar a la aplicación
Coste	El coste de migración a servicios Cloud podría ser superior al mantenimiento de la infraestructura que se está utilizando, por lo que se descarta. Contratar un sistema de protección anti DDoS supondría depender de terceros y realizar un desembolso de dinero, sin embargo es un coste asumible

Tabla 5.7: Salvaguardas para un ataque DDoS.

activo	amenaza	probabilidad	Degradación			Valoración (calculado)			Impacto (calculado)			Riesgo (calculado)			Efecto salvaguardas (agregadas)			Riesgo residual (calculado)		
			dimensión			dimensión			dimensión			dimensión			dimensión			dimensión		
			D	I	C	D	I	C	D	I	C	D	I	C	D	I	C	D	I	C
Servicio de visualización de simulaciones	Ataque DdoS	Normal	_100%	_1%	_1%	Muy alto	Bajo	Bajo	MA	MB	MB	MA	MB	MB	minimizan riesgo	na	na	B	-	-
Servicio de creación de simulaciones	Ataque DdoS	Normal	_100%	_1%	_1%	Muy alto	Bajo	Bajo	MA	MB	MB	MA	MB	MB	minimizan riesgo	na	na	B	-	-
Base de datos	Ataque DdoS	Normal	_1%	_1%	_1%	Muy alto	Medio	Muy alto	M	MB	M	M	MB	M	na	na	na	-	-	-
Software: Aplicación Web	Ataque DdoS	Normal	_100%	_1%	_1%	Alto	Medio	Bajo	A	MB	MB	A	MB	MB	minimizan riesgo	na	na	B	-	-
Software: SUMO	Ataque DdoS	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Software: VEIN	Ataque DdoS	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Software: SGBD	Ataque DdoS	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Hardware: servidor AppWeb	Ataque DdoS	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Hardware: servidor BD	Ataque DdoS	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Sala de servidores	Ataque DdoS	Normal	_1%	_1%	_1%	Bajo	Bajo	Bajo	MB	MB	MB	MB	MB	MB	na	na	na	-	-	-

Figura 5.6: Resumen de la amenaza de ataque de denegación de servicio.

5.2. Plan de mantenimiento

Dentro del plan de mantenimiento se especifican tres tareas a realizar: la realización de copias de seguridad de la base de datos, la comprobación de actualizaciones del software utilizado y el mantenimiento del servidor web.

La primera de ellas tiene como objetivo evitar la pérdida de información, y es tarea del administrador de la base de datos. Se ha decidido que la periodicidad de dichas copias sea semanal y que existan 2 réplicas, es decir, se tendrán dos servidores distintos a los que se copiará la información de la base de datos. Uno de estos servidores se ubicará en la misma sala en la que se encuentra el servidor de la base de datos principal, mientras que el otro se ubicará en otro edificio, para así evitar que accidentes de causa mayor (como un incendio) provoquen la pérdida total de la información.

Respecto a la segunda tarea, esta está enfocada sobretodo en las actualizaciones del software SUMO y VEIN. En caso de haber disponible alguna actualización se tendría que valorar si es conveniente su instalación, evaluando si aporta alguna característica interesante respecto a las versiones que actualmente se están utilizando. Por otro lado, las actualizaciones de seguridad habría que instalarlas tan pronto como fuera posible, para evitar tener vulnerabilidades en el sistema.

Finalmente, la tercera tarea tiene como objetivo lograr que las caídas del sistemas sean mínimas. Para ello, el administrador del servidor tendrá que comprobar semanalmente el estado de dicho servidor, con el objetivo de detectar posibles intrusiones y ataques al servidor.

Capítulo 6

Conclusiones

En este último capítulo se dará la opinión personal sobre el proyecto. Además, se expondrán los esfuerzos invertidos y las perspectivas futuras.

6.1. Evaluación personal

Una vez finalizado el trabajo y evaluado lo realizado, valoro la experiencia como muy satisfactoria. En primer lugar, he sentido que hay una notable diferencia entre este trabajo y cualquiera de los que he realizado durante el grado, puesto que al encontrarse englobado dentro del proyecto TRAFAIR toda decisión ha estado mucho más medida y con vistas a ser útil dentro del proyecto. No solo eso, sino que incluso ha sido necesario colaborar con otras personas que incluso son ajenas al mundo de la informática.

Además cabe destacar el hecho de haber aprendido sobre herramientas de simulación de tráfico, cuyo funcionamiento desconocía. Por otro lado, ha sido interesante adquirir conocimientos medioambientales y sobre la contaminación producida por el tráfico.

Integrar herramientas existentes por no ser viable abordar un desarrollo desde cero ha sido más complejo de lo que esperaba inicialmente. La documentación disponible no estaba lo suficientemente detallada y en muchos casos fue necesario realizar pruebas para validar si el funcionamiento de las herramientas empleadas se ajustaba a lo especificado en la documentación.

En definitiva, ha sido una experiencia enriquecedora ya que se han utilizado herramientas y adquirido conocimientos que no se habían visto a lo largo del grado, y he podido aprender mucho colaborando en un proyecto real, interactuando con sus investigadores y siguiendo las orientaciones recibidas.

6.2. Comparación del sistema desarrollado con los desarrollados en Módena y Santiago

Inicialmente se deseaba comparar el sistema desarrollado con los sistemas desarrollados en las ciudades de Módena y Santiago de Compostela. Sin embargo, no ha sido posible puesto que en ninguna de dichas ciudades ha finalizado el desarrollo del sistema. Aún así, se conocen algunas características de dichos sistemas que se van a mencionar.

En la ciudad de Módena también se utiliza SUMO y VEIN. Además, el volumen de datos de tráfico del que se dispone es mayor que el de Zaragoza, ya que en Módena el número de estaciones permanentes instaladas es mayor que en Zaragoza.

En Santiago se va a implementar un sistema basado en redes neuronales para calcular los flujos de tráfico, en vez de utilizar SUMO. Además, se dispone de datos de tráfico en tiempo real, lo que permite realizar monitorización de tráfico.

6.3. Metodología y esfuerzos invertidos

Para la realización del proyecto se ha seguido una metodología basada en el ciclo de Deming (Planificar-Hacer-Verificar-Actuar), para asegurar la calidad del producto final.

Respecto a los esfuerzos invertidos, el proyecto comenzó el 20 de mayo de 2019, justo al finalizar el curso académico, y su duración ha sido de 4 meses. En la figura 6.1 y en la tabla 6.1 se detallan los esfuerzos invertidos.

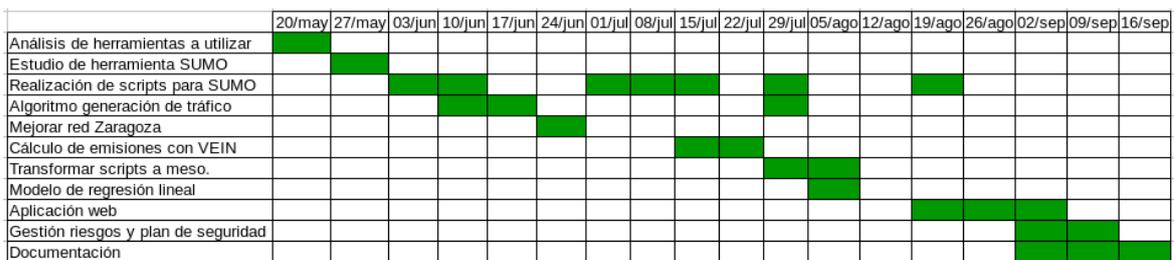


Figura 6.1: Diagrama de Gantt.

Tarea	Tiempo (h)
Análisis de herramientas a utilizar	10
Estudio de herramienta SUMO	10
Realización de scripts para SUMO	100
Algoritmo generación de tráfico	20
Mejorar red Zaragoza	25
Cálculo de emisiones con VEIN	40
Transformar scripts a modelo mesoscópico	10
Modelo de regresión lineal	10
Aplicación web	60
Gestión riesgos y plan de seguridad	15
Documentación	50
Total	350

Tabla 6.1: Horas invertidas en el proyecto.

6.4. Trabajo futuro

Al encontrarse el presente trabajo dentro del proyecto TRAFAIR, hay varias líneas de trabajo que se pueden abordar. Como trabajo futuro a desarrollar, se propone la mejora del modelo de tráfico. Actualmente se están utilizando datos de 46 estaciones permanentes distintas, pero además se dispone de datos de otras estaciones llamadas temporales, que cumplen la misma función que las permanentes con la diferencia de que son movidas de ubicación periódicamente. Se considera que, teniendo en cuenta los datos de las estaciones temporales, se puede obtener un modelo de tráfico más acorde a la realidad.

Otra posible línea de trabajo sería la mejora de la aplicación web. Por ejemplo, sería interesante que el usuario fuera capaz de indicar situaciones hipotéticas para las simulaciones, como cortes de tráfico o restricción de acceso de determinados tipos de vehículos a la ciudad, para así ver cómo afectan estas situaciones a los flujos de tráfico y a la contaminación. Por último, también se considera de interés implementar cambios en la flota de vehículos, pues los datos de los que se dispone actualmente son incompletos.

Bibliografía

- [1] Mordechai (Muki) Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, October 2008.
- [2] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, November 2018.
- [3] S. Ibarra-Espinosa, R. Ynoue, S. O’Sullivan, E. Pebesma, M. D. F. Andrade, and M. Osses. Vein v0.2.2: an r package for bottom–up vehicular emissions inventories. *Geoscientific Model Development*, 11(6):2209–2229, 2018.
- [4] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.
- [5] Jakob Erdmann. Online-kalibrierung einer mikroskopischen verkehrssimulation. In *ViMOS 2012*, November 2012.
- [6] Nils Gustaf Eissfeldt. *Vehicle-based modelling of traffic . Theory and application to environmental impact modelling*. PhD thesis, Universität zu Köln, 2004.
- [7] Debashish Chowdhury, Ludger Santen, and Andreas Schadschneider. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, 329, 07 2000.
- [8] *Magerit version 1. 0. Metodología de analisis y gestion de riesgos de los sistemas de informacion. 5, Guia para responsables del dominio protegible*. Manuales. Administración general. Ministerio de Administraciones Públicas, 1997.

Anexos

Anexo A

SUMO

SUMO (Simulation of Urban MObility) es un paquete de código abierto de simulación de tráfico que permite simular cómo un conjunto de coches se mueven dentro de una red de carreteras. Esta simulación puede ser tanto microscópica como mesoscópica, según se desee. En una simulación microscópica los vehículos se definen explícitamente, es decir, cada vehículo tiene definida una ruta determinada y se mueve individualmente dentro de la red. En una simulación mesoscópica las rutas de los coches se definen de igual manera. Sin embargo, estos no se mueven individualmente dentro de la red, si no que cada calla (arista en SUMO) tiene definidas una o varias colas, de las cuales van entrando y saliendo coches en función de diversas características, como la saturación del tráfico o los semáforos cercanos.

Como características a destacar se pueden mencionar la posibilidad de definir diferentes tipos de vehículos con características diversas, definir rutas predeterminadas o aleatorias para los vehículos, ajustar el tráfico por un determinado punto al valor deseado, definir semáforos y reglas de paso en las intersecciones, definición de transporte público... La definición de todos estos elementos está basada en ficheros XML. En definitiva, la herramienta ofrece una serie de características que permiten realizar simulaciones muy cercanas a la realidad.

Respecto a las simulaciones, estas se dividen en pasos, correspondiéndose un paso con la simulación de un segundo en la realidad. Por ejemplo, si se desea realizar una simulación de una hora real, la simulación contará con 3600 pasos.

Además del propio simulador, el paquete incluye otras herramientas de interés:

- SUMO-GUI. Interfaz gráfica que permite ver cómo van moviéndose los coches por la red a medida que avanza el tiempo en la simulación.
- NETEDIT. Interfaz gráfica que permite modificar de forma sencilla una red.
- NETCONVERT. Transforma redes en otros formatos al formato de SUMO. Entre otros, permite transformar redes de OpenStreetMap.

Asimismo, el paquete también viene acompañado de una serie de scripts escritos en Python para facilitar el uso de estas herramientas.

Finalmente, cabe destacar que la herramienta SUMO permite utilizar dos modelos de tráfico distintos para realizar simulaciones: un modelo mesoscópico y un modelo microscópico. En el modelo microscópico cada vehículo se mueve independientemente por la red y decide su velocidad, cuando cambia de carril, cuando detenerse... en función de los elementos que va encontrándose en su entorno, como semáforos, otros coches, intersecciones, etc. Por otro lado, en el modelo mesoscópico cada uno de los segmentos de la red tiene definidas una o varias colas (según el número de carriles que tenga) de las que van insertando y sacando vehículos en función de varios parámetros: longitud de la vía, velocidad permitida en la vía, congestión de la vía, congestión de vías cercanas, estado de semáforos, etc. Es decir, el movimiento de los coches dentro de la red se basa en ir pasando de una cola a otra. La idea inicial era utilizar el modelo microscópico, sin embargo finalmente se ha decidido utilizar el modelo mesoscópico ya que se han obtenido mejores resultados que con el modelo microscópico, además de no haber producido tantos problemas como el modelo microscópico.

Anexo B

Tipos de vía

En este anexo se especifican los tipos de vías que hay en OSM. El peso se corresponde con un valor que es asignado a las aristas pertenecientes a ese tipo de vía.

Tipo de vía	Peso
motorway	28
trunk	13
primary	12
secondary	11
tertiary	10
motorway_link	9
trunk_link	8
primary_link	7
secondary_link	6
tertiary_link	5
unclassified	4
residential	3
living_street	2
track	1
unsurfaced	1
service	1

Tabla B.1: Peso para cada uno de los tipos de vía.

Anexo C

Correspondencias de VEIN con la información de la base de datos

En este anexo se especifican las correspondencias de los datos sobre la flota de vehículos de Zaragoza con sus valores equivalentes dentro del paquete VEIN.

BD	VEIN
passenger_car	PC
motorcycle	Motorcycle
moped	Moped
light_commercial_vehicle	LCV
bus	Coach
heavy_duty_truck	Trucks

Tabla C.1: Correspondencias de la clase de vehículo.

BD	VEIN
lpg	LPG
diesel	D
petrol	G

Tabla C.2: Correspondencias del carburante.

BD	VEIN
0.05	<=50
0.125	>=50
0.25	<=250
0.5	250_750
0.75	250_750
1.0	>=750
1.2	<=1400
1.6	>1400
2.0	1400_2000
2.5	>2000

Tabla C.3: Correspondencias de la cilindrada.

BD	VEIN
conventional	PRE
ece 15/04	PRE
euro 1	I
euro 2	II
euro 3	III
euro 4	IV
euro 5	V
euro 6	VI
euro 6 up to 2016	VIc
euro 6 up to 2017	VIc
euro 6 2017-2019	VIc

Tabla C.4: Correspondencias del estándar de emisión.

Anexo D

Mapa de navegación de TraFlow

En la figura D.1 se muestra el mapa de navegación de la aplicación, que ilustra el flujo entre ventanas que puede suceder por el uso de la aplicación. Cada uno de los números indicados en el mapa se corresponde con una de las figuras D.2, D.3, D.4, D.5, D.6, D.7, D.8, D.9 o D.10.

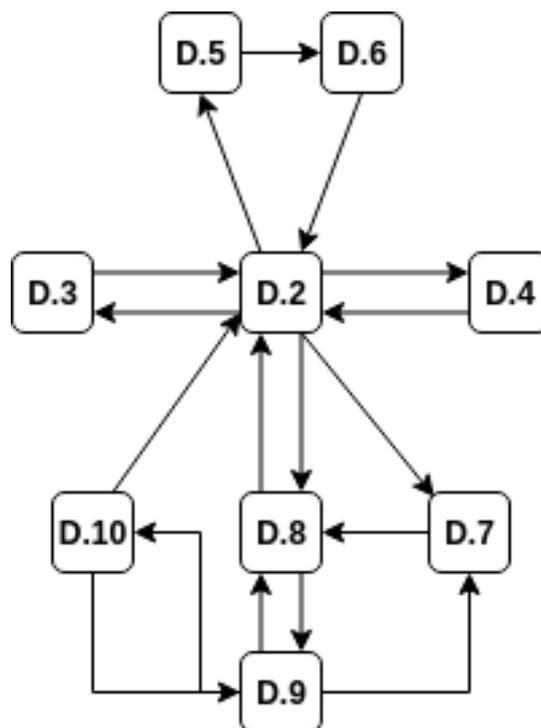


Figura D.1: Mapa de navegación de la aplicación.

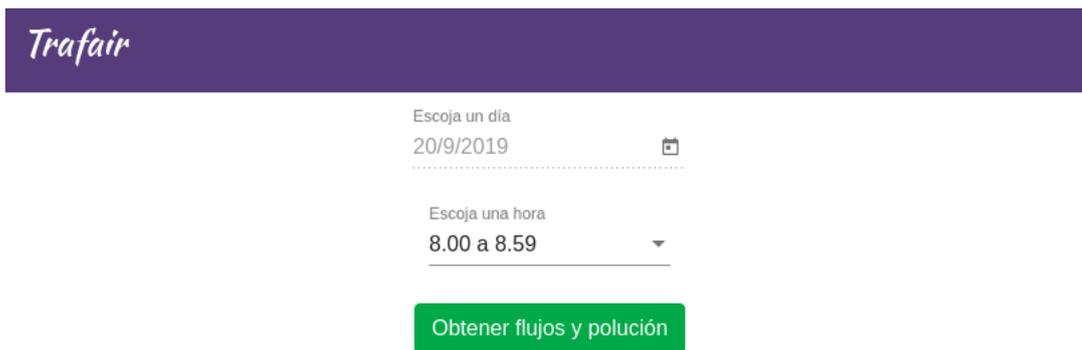


Figura D.2: Ventana inicial de la aplicación.

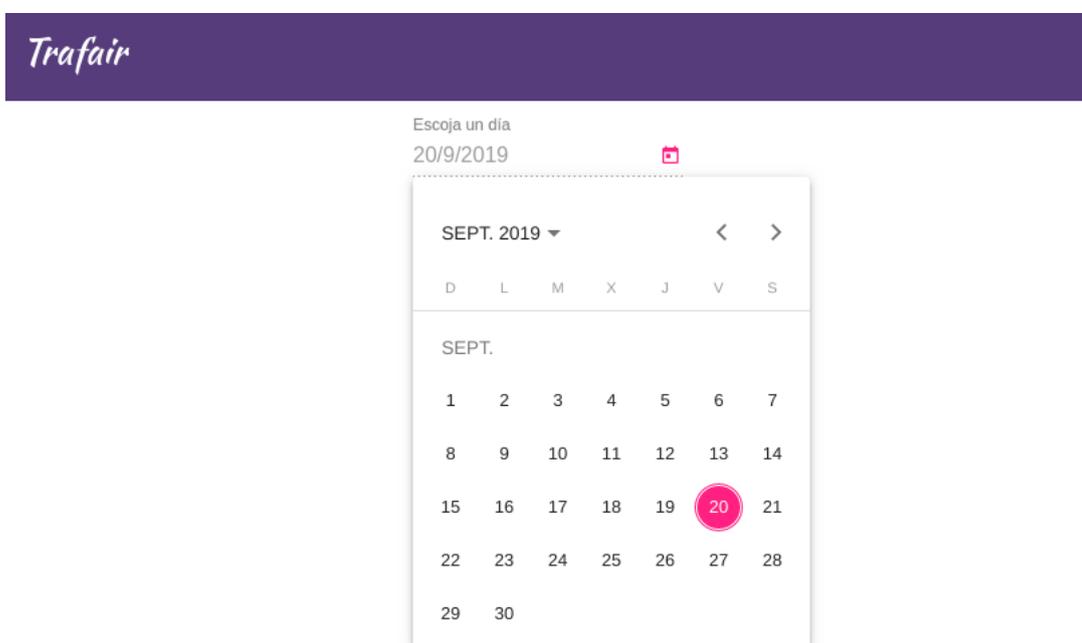


Figura D.3: Calendario mostrado en la aplicación.

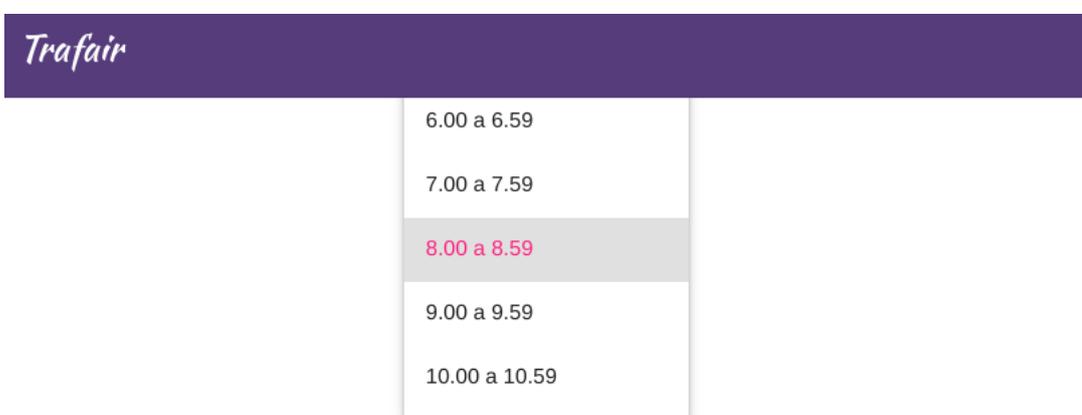


Figura D.4: Elección de hora.

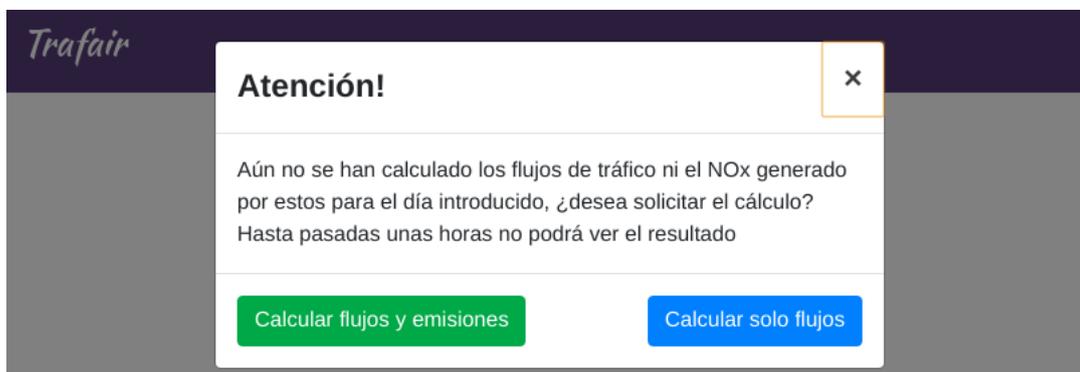


Figura D.5: Aviso de que no hay datos.

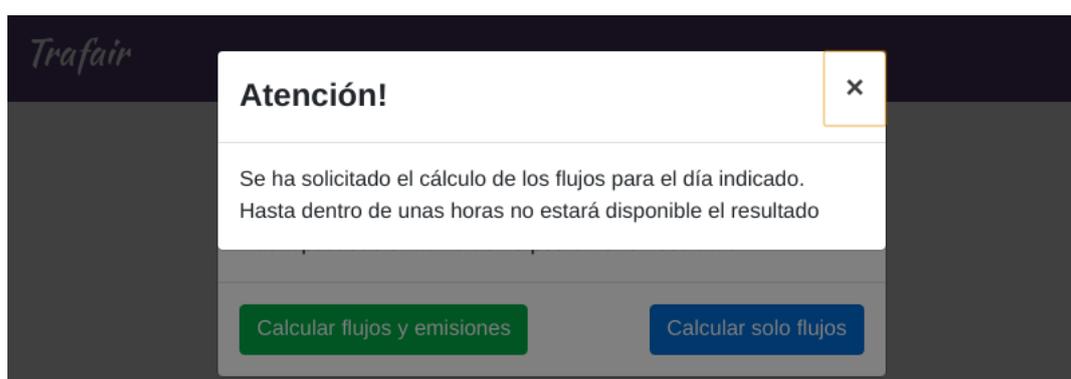


Figura D.6: Aviso de que se ha lanzado simulación.



Figura D.7: Aviso de que se solo hay datos de flujos.

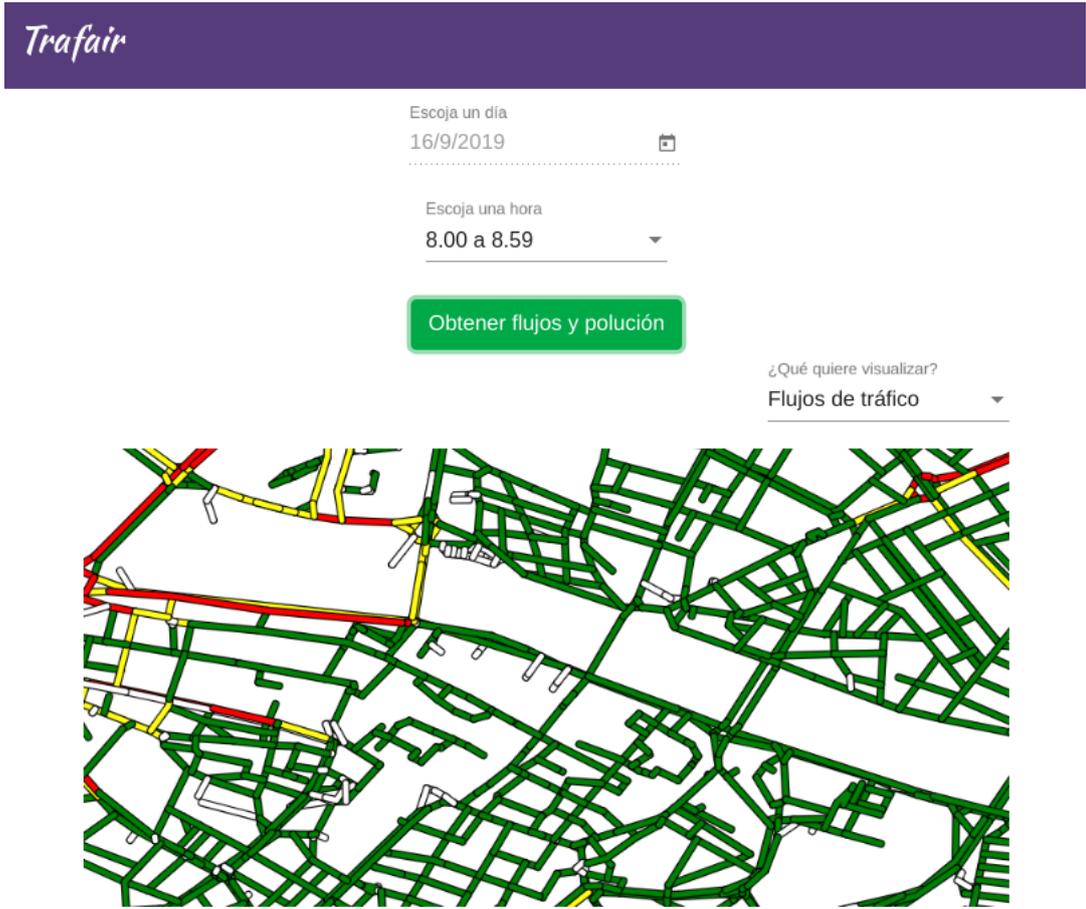


Figura D.8: Mapa mostrando flujos por colores.

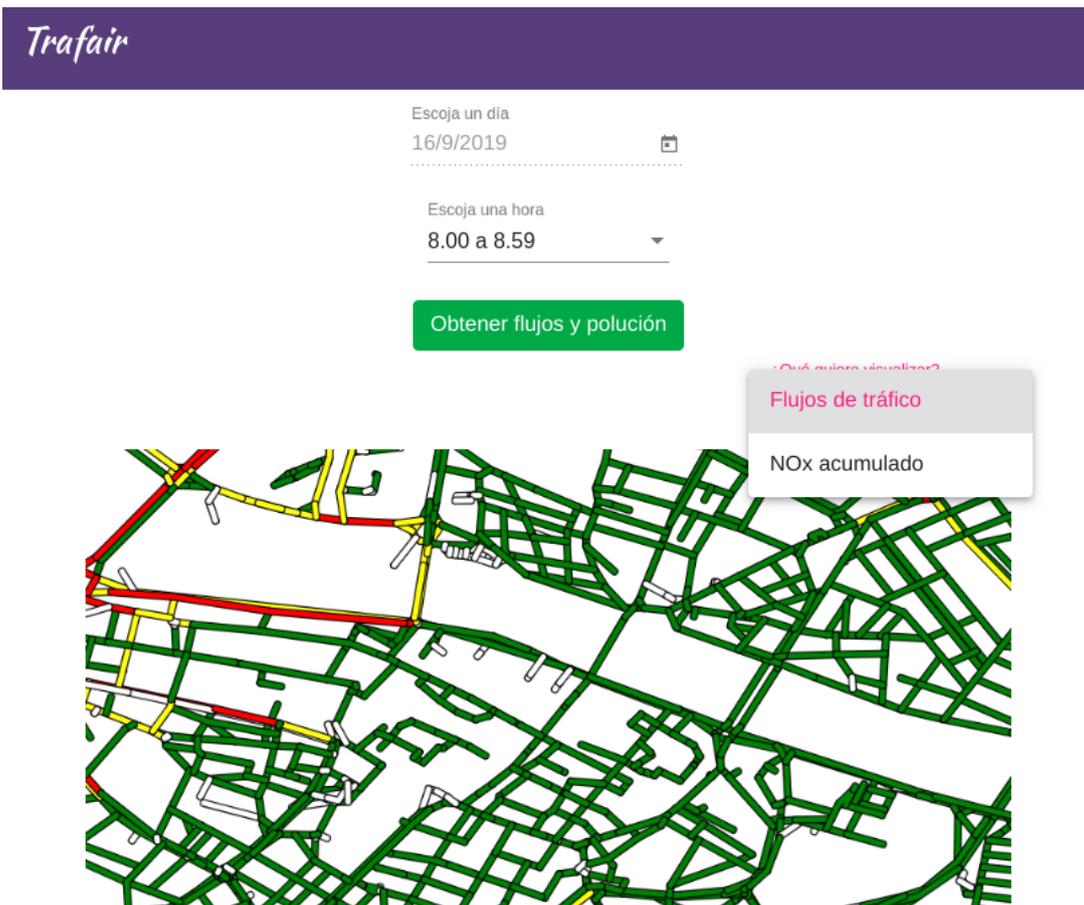


Figura D.9: Opción de cambiar de mapa.

Trafaïr

Escoja un día

16/9/2019



Escoja una hora

8.00 a 8.59



Obtener flujos y polución

¿Qué quiere visualizar?

NOx acumulado



Figura D.10: Mapa mostrando acumulación de NOx por colores.

Anexo E

Magerit detallado

En este anexo se detallan las amenazas identificadas (ver tablas E.1, E.1, E.2 y E.4). Además, se aporta la valoración del impacto, riesgo y riesgo residual respecto de las amenazas inundación, fallo de base de datos e intrusión (ver figuras E.2, E.1 y E.3).

Inundación en la sala de servidores	
Tipos de activos a los que afecta	Dimensión
[HW] equipos informáticos	[D] Disponibilidad
[S] servicios	[I] Integridad
[L] instalaciones	
Descripción	Origen
Posibilidad de que una inundación destruya los recursos HW y SW necesarios para acceder a los servicios	Accidental

Tabla E.1: Amenaza de inundación.

Avería en el servidor de BD	
Tipos de activos a los que afecta	Dimensión
[HW] servidor	[D] Disponibilidad
[SW] aplicación web	[I] Integridad
[S] Servicios	
Descripción	Origen
Fallo en el servidor de BD. Su fallo puede deberse a la rotura de algún componente hardware, a algún fallo del software instalado o por algún tipo de error de configuración	Accidental
	Humano

Tabla E.2: Amenaza de fallo en BD.

Intrusión en la base de datos	
Tipos de activos a los que afecta	Dimensión
[HW] servidor	[D] Disponibilidad
[SW] aplicación web	[I] Integridad
[S] Servicios	[C] Confidencialidad
Descripción	Origen
Una persona no autorizada accede al servidor de bases de datos, bien sea porque ha conseguido la clave del administrador o porque ha aprovechado una vulnerabilidad	Humano

Tabla E.3: Amenaza de intrusión.

Ataque de denegación de servicio	
Tipos de activos a los que afecta	Dimensión
[S] servicios	[D] Disponibilidad
	[I] Integridad
	[C] Confidencialidad
Descripción	Origen
Los atacantes intentan que los usuarios no puedan acceder a la aplicación haciendo peticiones masivas	Humano

Tabla E.4: Amenaza de denegación de servicio.

activo	amenaza	probabilidad	Degradación			Valoración (calculado)			Impacto (calculado)			Riesgo (calculado)			Efecto salvaguardas (agregadas)			Riesgo residual (calculado)		
			dimensión			dimensión			dimensión			dimensión			dimensión			dimensión		
			D	I	C	D	I	C	D	I	C	D	I	C	D	I	C	D	I	C
Servicio de visualización de simulaciones	Intrusión	Infrecuente	_1%	_10%	_100%	Muy alto	Bajo	Bajo	M	MB	B	M	MB	B	na	minimizan riesgo	minimizan riesgo	-	MB	MB
Servicio de creación de simulaciones	Intrusión	Infrecuente	_1%	_10%	_100%	Muy alto	Bajo	Bajo	M	MB	B	M	MB	B	na	minimizan riesgo	minimizan riesgo	-	MB	MB
Base de datos	Intrusión	Infrecuente	_1%	_100%	_100%	Muy alto	Medio	Muy alto	M	M	MA	M	M	MA	na	minimizan riesgo	minimizan riesgo	-	MB	B
Software: Aplicación Web	Intrusión	Infrecuente	_1%	_10%	_100%	Alto	Medio	Bajo	B	B	B	B	B	B	na	minimizan riesgo	minimizan riesgo	-	MB	MB
Software: SUMO	Intrusión	Infrecuente	_1%	_10%	_100%	Alto	Bajo	Bajo	B	MB	B	B	MB	B	na	minimizan riesgo	minimizan riesgo	-	MB	MB
Software: VEIN	Intrusión	Infrecuente	_1%	_10%	_100%	Alto	Bajo	Bajo	B	MB	B	B	MB	B	na	minimizan riesgo	minimizan riesgo	-	MB	MB
Software: SGBD	Intrusión	Infrecuente	_1%	_10%	_100%	Alto	Bajo	Bajo	B	MB	B	B	MB	B	na	minimizan riesgo	minimizan riesgo	-	MB	MB
Hardware: servidor AppWeb	Intrusión	Infrecuente	_1%	_1%	_100%	Alto	Bajo	Bajo	B	MB	B	B	MB	B	na	na	minimizan riesgo	-	-	MB
Hardware: servidor BD	Intrusión	Infrecuente	_1%	_1%	_100%	Alto	Bajo	Bajo	B	MB	B	B	MB	B	na	na	minimizan riesgo	-	-	MB
Sala de servidores	Intrusión	Infrecuente	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-

Figura E.1: Resumen de la amenaza de intrusión.

activo	amenaza	probabilidad	Degradación			Valoración (calculado)			Impacto (calculado)			Riesgo (calculado)			Efecto salvaguardas (agregadas)			Riesgo residual (calculado)		
			dimensión			dimensión			dimensión			dimensión			dimensión			dimensión		
			D	I	C	D	I	C	D	I	C	D	I	C	D	I	C	D	I	C
Servicio de visualización de simulaciones	Inundación	Muy infrecuente	100%	1%	1%	Muy alto	Bajo	Bajo	MA	MB	MB	A	MB	MB	minimizan riesgo	na	na	B	-	-
Servicio de creación de simulaciones	Inundación	Muy infrecuente	100%	1%	1%	Muy alto	Bajo	Bajo	MA	MB	MB	A	MB	MB	minimizan riesgo	na	na	B	-	-
Base de datos	Inundación	Muy infrecuente	100%	100%	1%	Muy alto	Medio	Muy a	MA	M	M	A	B	B	minimizan riesgo	minimizan riesgo	na	B	MB	-
Software: Aplicación Web	Inundación	Muy infrecuente	1%	1%	1%	Alto	Medio	Bajo	B	MB	MB	MB	MB	MB	na	na	na	-	-	-
Software: SUMO	Inundación	Muy infrecuente	1%	1%	1%	Alto	Bajo	Bajo	B	MB	MB	MB	MB	MB	na	na	na	-	-	-
Software: VEIN	Inundación	Muy infrecuente	1%	1%	1%	Alto	Bajo	Bajo	B	MB	MB	MB	MB	MB	na	na	na	-	-	-
Software: SGBD	Inundación	Muy infrecuente	1%	1%	1%	Alto	Bajo	Bajo	B	MB	MB	MB	MB	MB	na	na	na	-	-	-
Hardware: servidor AppWeb	Inundación	Muy infrecuente	100%	1%	1%	Alto	Bajo	Bajo	A	MB	MB	M	MB	MB	minimizan riesgo	na	na	MB	-	-
Hardware: servidor BD	Inundación	Muy infrecuente	100%	1%	1%	Alto	Bajo	Bajo	A	MB	MB	M	MB	MB	minimizan riesgo	na	na	MB	-	-
Sala de servidores	Inundación	Muy infrecuente	100%	1%	1%	Alto	Bajo	Bajo	A	MB	MB	M	MB	MB	minimizan riesgo	na	na	MB	-	-

Figura E.2: Resumen de la amenaza de inundación.

activo	amenaza	probabilidad	Degradación			Valoración (calculado)			Impacto (calculado)			Riesgo (calculado)			Efecto salvaguardas (agregadas)			Riesgo residual (calculado)		
			dimensión			dimensión			dimensión			dimensión			dimensión			dimensión		
			D	I	C	D	I	C	D	I	C	D	I	C	D	I	C	D	I	C
Servicio de visualización de simulaciones	Avería BD	Normal	_10%	_1%	_1%	Muy alto	Bajo	Bajo	A	MB	MB	A	MB	MB	minimizan riesgo	na	na	B	-	-
Servicio de creación de simulaciones	Avería BD	Normal	_10%	_1%	_1%	Muy alto	Bajo	Bajo	A	MB	MB	A	MB	MB	minimizan riesgo	na	na	B	-	-
Base de datos	Avería BD	Normal	_100%	_100%	_1%	Muy alto	Medio	Muy a	MA	M	M	MA	M	M	minimizan riesgo	minimizan riesgo	na	B	MB	-
Software: Aplicación Web	Avería BD	Normal	_1%	_1%	_1%	Alto	Medio	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Software: SUMO	Avería BD	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Software: VEIN	Avería BD	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Software: SGBD	Avería BD	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Hardware: servidor AppWeb	Avería BD	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Hardware: servidor BD	Avería BD	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-
Sala de servidores	Avería BD	Normal	_1%	_1%	_1%	Alto	Bajo	Bajo	B	MB	MB	B	MB	MB	na	na	na	-	-	-

Figura E.3: Resumen de la amenaza de avería.

Anexo F

Esquema Entidad-Relación de la base de datos de TRAFAIR

En las figuras se detalla el esquema utilizado en la base de datos de TRAFAIR.

Vehicle fleet composition

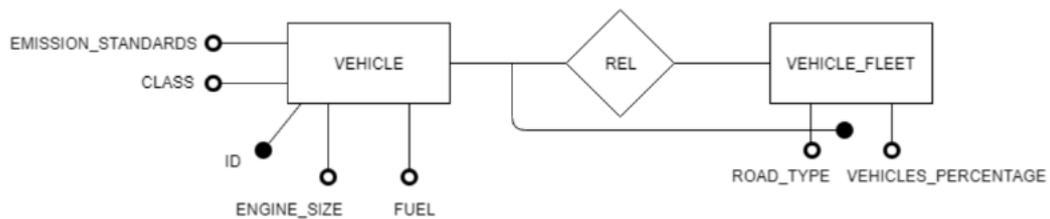


Figura F.1: Tipo de entidad vehículo.

Emission factors for vehicles

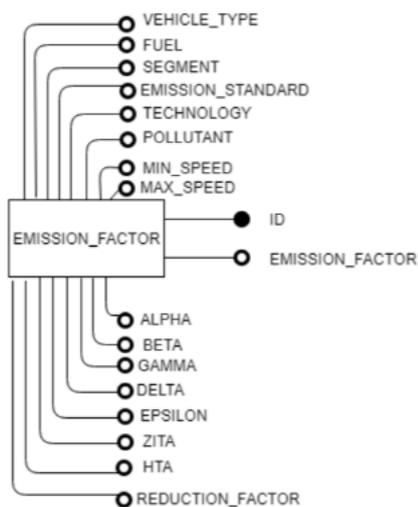


Figura F.2: Tipo de entidad factores de emisión.

Road network and real-time traffic flow

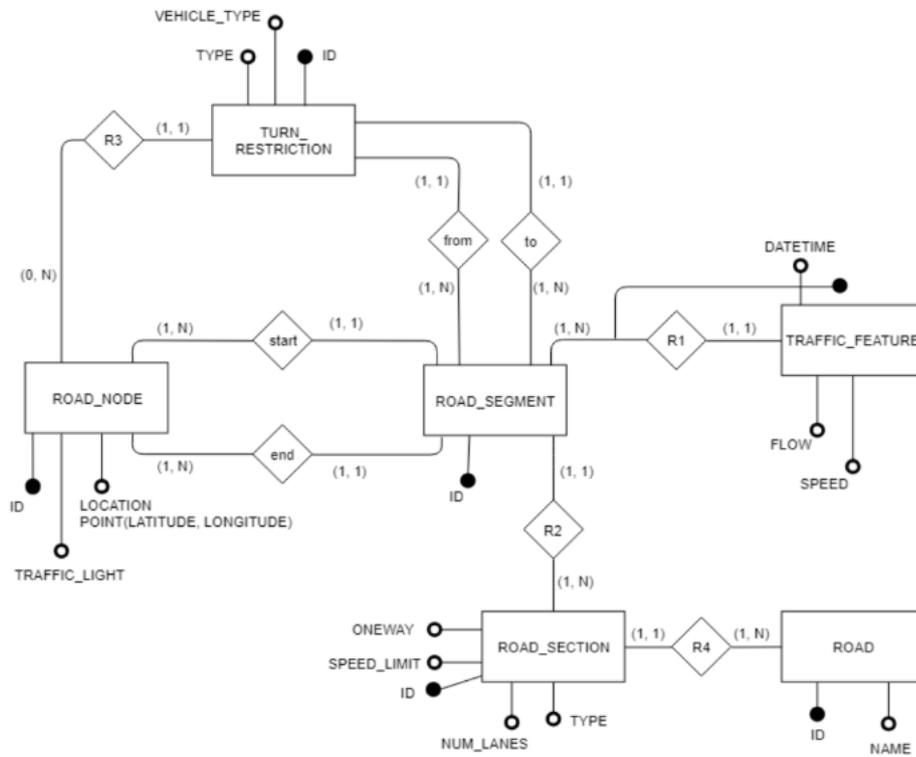


Figura F.3: Tipo de entidad road.

Emission factors for vehicles

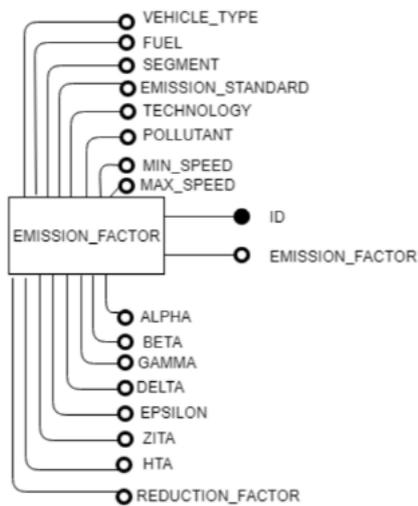


Figura F.4: Tipo de entidad sensor.