



UNIVERSIDAD
DE MÁLAGA



TRABAJO FIN DE GRADO

**INFRAESTRUCTURAS DE CARGA
DE VEHÍCULOS ELÉCTRICOS.
DINÁMICA DE TRÁFICO URBANO CON ETAPAS.**

GRADO EN INGENIERÍA DE LA ENERGÍA

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE MÁLAGA

Autor: Santiago Torres Cardetti

Tutores: Francisco Jesús Muñoz Gutiérrez

Alejandro Rodríguez Gómez

Área de conocimiento: Ingeniería Eléctrica

Málaga, junio de 2025

INFRAESTRUCTURAS DE CARGA DE VEHÍCULOS ELÉCTRICOS. DINÁMICA DE TRÁFICO URBANO CON ETAPAS.

Resumen

Este Trabajo de Fin de Grado analiza el comportamiento de los conductores de vehículos eléctricos, simulando una red de transporte mediante herramientas de programación (Python®), donde una serie de vehículos eléctricos realizan distintos viajes, parando para recargar si es necesario en las distintas estaciones de carga, y simulación (AimSun Next®), donde se simula un flujo de tráfico completo en una zona real.

El comportamiento de dichos conductores difiere del de los conductores de vehículos de combustión tradicionales, ya que existe el estado de carga o SoC (en inglés State of Charge) que el conductor debe controlar en todo momento para seguir su ruta, teniendo en mente el tiempo que le tomará recargar dependiendo del tipo de cargador que esté disponible en la estación de carga (EdC) seleccionada para la parada.

Para ello y con el fin de obtener una simulación lo más exacta y afín a la realidad, se diseñan: un algoritmo que calcule y actualice el SoC en cada punto de la ruta, se utilizan EdC existentes y se utilizan flujos de tráfico similares a la realidad, pudiendo realizar estimaciones de las localizaciones óptimas donde deberían existir una EdC.

En este caso, las simulaciones se enmarcan en Málaga, en dos zonas diferentes y con dos estados iniciales distintos.

Palabras clave: Vehículo eléctrico, estado de carga, estación de carga, dinámica de tráfico.

ELECTRIC VEHICLE CHARGING INFRASTRUCTURE. URBAN TRAFFIC DYNAMICS WITH STAGES.

Abstract

This Bachelor's Thesis analyzes the behavior of electric vehicle drivers by simulating a transportation network using programming tools (Python®), where a series of electric vehicles make various trips, stopping to recharge if necessary at different charging stations, and simulation software (AimSun Next®), where a complete traffic flow is simulated in a real area.

The behavior of these drivers differs from that of traditional internal combustion engine vehicle drivers, due to the existence of the State of Charge (SoC), which the driver must monitor at all times in order to follow their route—keeping in mind the time required to recharge, depending on the type of charger available at the selected charging station (CS).

To achieve a simulation as accurate and realistic as possible, the following are implemented: an algorithm that calculates and updates the SoC at each point along the route, real existing CSs, and traffic flows similar to real-life conditions, allowing for estimations of the optimal locations where CSs should be placed.

In this case, the simulations are set in Málaga, covering two different areas and two distinct initial conditions.

Keywords: Electric vehicle, state of charge, charging station, traffic dynamics.

A Lucía, por estar a mi lado día a día.

A mi Madre, por apoyarme pase lo que pase.

A mi Padre, por motivarme a seguir en todo momento.

Agradecimientos

Primero me gustaría agradecer a mis tutores, Francisco y Alejandro, que desde el momento en el que les escribí para realizar este TFG, se pusieron manos a la obra para que fuese lo mejor posible, ofreciéndome las mejores herramientas y métodos desde el día uno.

Continuando, quiero agradecer desde el corazón a mi pareja, Lucía, que me ha acompañado no solo durante la realización del proyecto, sino desde el principio. Pese a tener complicaciones para abordar todos los retos que este TFG me ha supuesto, siempre he podido contar contigo como estudiante de ingeniería y como compañera.

Me gustaría agradecer especialmente a mis padres y abuelos, sois personas fundamentales en mi vida, sin vuestro apoyo y ánimo jamás podría haber completado este proyecto de la manera en la que lo he hecho.

También quiero agradecer a todos los familiares y amigos con los que he tenido la oportunidad de hablar sobre este proyecto y de los cuales he aprendido desde un punto de vista distinto al mío.

Finalmente, deciros a todos que me esforzaré día a día para hacer que estéis orgullosos, porque siento que es la única manera de agradecer de verdad todo lo que habéis hecho y seguís haciendo por mí.

Índice

1. Introducción	1
1.1. Conceptos Preliminares	1
1.2. Alcance del Proyecto	2
1.3. Motivación	2
1.4. Estado del Arte	3
2. Metodología y programas	7
2.1. Python®, base de datos, mapeado y rutas	7
2.1.1. Librerías utilizadas	8
2.2. AimSun Next®, métodos y restricciones	9
2.2.1. Métodos de simulación	9
2.2.2. Restricciones	11
3. Casos de estudio y resultados	13
3.1. Caso 1: Simulación con Python®	13
3.1.1. Configuración inicial	13
3.1.2. Algoritmo de cálculo del SoC	15
3.1.3. Bucle de cálculo de rutas	17
3.1.4. Resultados obtenidos con Python®	18
3.2. Caso 2: Simulación con AimSun Next®	23
3.2.1. Base de la simulación	23
3.2.1.1. Zona 1: Málaga centro	24
3.2.1.2. Zona 2: Ciudad Jardín norte	25

3.2.1.3.	Zona 3: Puerta Blanca	26
3.2.1.4.	Zona 4: El Viso/Teatinos	27
3.2.2.	Adaptación e implantación de la zona	28
3.2.3.	Recreación de las vías y objetos de tráfico	29
3.2.3.1.	Vías	29
3.2.3.2.	Nodos y rotondas	30
3.2.3.3.	Señales de tráfico	31
3.2.4.	Cuerpo de la simulación	32
3.2.4.1.	Centroides	32
3.2.4.2.	Matriz Origen-Destino	34
3.2.4.3.	Porcentaje de giro	35
3.3.	Resultados de la simulación	36
4.	Conclusiones, propuestas de mejora y aplicación práctica	43
4.1.	Conclusiones y propuestas de mejora	43
4.1.1.	Influencia del (SoC) en el comportamiento del conductor	44
4.1.2.	Algoritmo de consumo personalizado por características del vehículo	44
4.1.3.	Optimización de rutas con recarga intermedia	45
4.1.4.	Localización estratégica de EdC según densidad y flujo de tráfico	45
4.1.5.	Limitación por falta de datos reales	45
4.1.6.	Combinación de herramientas de simulación como base metodológica	46
4.2.	Aplicación práctica propuesta	46
Referencias		47
Anexos		49
Anexo A. Resultados complementarios		49

Índice de Tablas

1.	Resumen de vehículos y consumo	22
A.1.	Resumen completo de resultados de la simulación con AimSun	49

Índice de Ilustraciones

1.1.	Modelo Ohio Baker Electric V Victoria	4
1.2.	Composición de las baterías de ion de litio y estado sólido.	5
1.3.	Programas utilizados para la realización de las simulaciones.	6
2.1.	Ejemplo de matriz OD	10
2.2.	Web de OpenStreetMap y ortofoto de la zona.	11
3.1.	Librerías utilizadas.	14
3.2.	Estaciones de carga seleccionadas.	15
3.3.	Algoritmo de cálculo del SoC.	16
3.4.	Rama de transformación de coordenadas a nodos.	17
3.5.	Rama de cálculo del primer tramo de la ruta.	17
3.6.	Rama de cálculo del segundo tramo de la ruta.	18
3.7.	Unión de tramos y creación de la ruta completa.	18
3.8.	Ejemplo de mensaje impreso por pantalla de cada ruta.	18
3.9.	Mapa con las 20 rutas.	19
3.10.	Kilómetros recorridos según el vehículo.	20
3.11.	Estado del SoC en momentos parciales de la ruta.	21
3.12.	Resultados del algoritmo según el SoC final.	21
3.13.	Ortofoto de la zona de Málaga centro.	24
3.14.	Ortofoto de la zona de Ciudad Jardín norte.	25
3.15.	Ortofoto de la zona de Puerta Blanca.	26
3.16.	Ortofoto de la zona de El Viso/Teatinos.	27
3.17.	Implementación de la zona escogida en AimSun Next®.	28

3.18.	Esquema de creación de vías y sus posibles modificaciones.	29
3.19.	Túnel a 5 metros de diferencia con el nivel del resto de las vías.	30
3.20.	Rotonda con los nodos seleccionados.	30
3.21.	Modo de implementación de una rotonda.	31
3.22.	Distintas señales recreadas de la realidad.	32
3.23.	Centroides implementados en la simulación.	33
3.24.	Matriz OD Total de todos los vehículos.	35
3.25.	Implementación de porcentajes.	36
3.26.	Inicialización de la demanda de tráfico en la simulación.	36
3.27.	Cola media de vehículos en las vías durante el intervalo de la simulación. . . .	37
3.28.	Densidad de vehículos en las vías durante el intervalo de la simulación.	38
3.29.	Densidad de vehículos en las vías durante el intervalo de la simulación.	39
3.30.	Detalle de flujo de vehículos en diferentes áreas.	40
3.31.	Velocidad y distancia recorrida por los vehículos.	41

Capítulo 1

Introducción

1.1. Conceptos Preliminares

Antes de adentrarse en el desarrollo del presente Trabajo de Fin de Grado, es fundamental comprender una serie de conceptos clave que forman la base del estudio:

- **Vehículo Eléctrico (EV, Electric Vehicle, en inglés):** Automóvil propulsado por uno o más motores eléctricos, alimentados mediante baterías recargables. A diferencia de los vehículos de combustión interna, los EVs no emiten gases contaminantes directamente, lo que los convierte en una alternativa sostenible.
- **Estado de Carga (SoC, por sus siglas en inglés, State of charge):** Porcentaje de energía restante en la batería de un vehículo eléctrico. Este parámetro es esencial para planificar trayectos y determinar si es necesario recargar durante un desplazamiento.
- **Estación de Carga (EdC):** Infraestructura que permite la recarga de las baterías de los EVs. Su disponibilidad, localización y capacidad influyen directamente en la viabilidad del uso masivo de vehículos eléctricos.
- **Simulación Microscópica y Macroscópica de Tráfico:** Modelos utilizados para reproducir el comportamiento del tráfico urbano. En este trabajo se emplean herramientas como *Python®* y *AimSun Next®* para modelar, respectivamente, trayectos individuales y flujos de tráfico agregados.

1.2. Alcance del Proyecto

Este proyecto tiene como finalidad diseñar un sistema de simulación que represente de manera realista el comportamiento de los conductores de EVs en entornos urbanos, considerando la necesidad de recarga en trayectos de media distancia.

Para ello, se implementan dos tipos de simulaciones:

1. **Simulación individualizada de rutas con Python®**: se analizan 20 trayectos de EVs con distintas características, evaluando el impacto del SoC en la toma de decisiones de recarga.
2. **Simulación agregada de tráfico urbano con AimSun Next®**: centrada en una zona concreta de la ciudad de Málaga, con el objetivo de identificar áreas de alta densidad vehicular donde la instalación de EdC sería más eficiente.

El alcance abarca desde la selección y procesamiento de datos, la implementación de algoritmos personalizados, hasta el análisis de resultados orientado a la optimización de infraestructura de recarga.

1.3. Motivación

La motivación de este trabajo surge de la necesidad urgente de transformar el modelo de movilidad actual hacia uno más sostenible, ante la inminente escasez de combustibles fósiles y la crisis ambiental global. Aunque los EVs ofrecen una alternativa viable, presentan limitaciones que aún obstaculizan su adopción masiva, siendo la autonomía y la infraestructura de carga dos de los principales retos.

La escasez de EdC en zonas urbanas y la falta de planificación estratégica en su ubicación generan incertidumbre en los conductores, fenómeno conocido como *ansiedad de autonomía*. Este proyecto pretende contribuir al diseño de soluciones inteligentes que reduzcan dicha ansiedad, mediante un enfoque combinado de simulación individual y urbana, orientado a la toma de decisiones óptimas de recarga.

Además, el trabajo representa una oportunidad académica para combinar herramientas de programación, modelado de tráfico y análisis energético, aplicadas a un problema actual y con gran proyección futura.

1.4. Estado del Arte

“Los vehículos eléctricos se han convertido en la forma de arreglar parte de la brecha creada por el ser humano en la capa de ozono”, eso es lo que se proyectaba hace una década. Debido a la revolución del motor de combustión, el enfoque ambiental a sido relegado a un segundo plano, hasta que la situación se ha convertido en una crisis mundial. Teniendo en cuenta que las reservas de petróleo se estiman agotadas en 40 años [1] si se dejan de encontrar más fuentes (como está ocurriendo en la mayoría de los países exportadores), la búsqueda de una alternativa para la movilidad es un tema urgente.

Los EVs, desde su creación en 1830 (siendo EVs de batería no recargable) hasta el día de hoy, siempre han tenido la misma problemática y desventaja comparados con los vehículos de combustión, su autonomía.

A finales del siglo XIX y comienzos del siglo XX se empiezan a producir en masa las baterías recargables, siendo el EV una opción muy codiciada en las altas clases debido a su precio elevado.

El modelo más vendido fue el Ohio Baker Electric con baterías diseñadas por Thomas Edison (figura 1.1.a).



Nota. Reproducido de "1908 - Baker Electrics Victoria", por PlanetCarsZ, s.f..

Figura 1.1. Modelo Ohio Baker Electric V Victoria

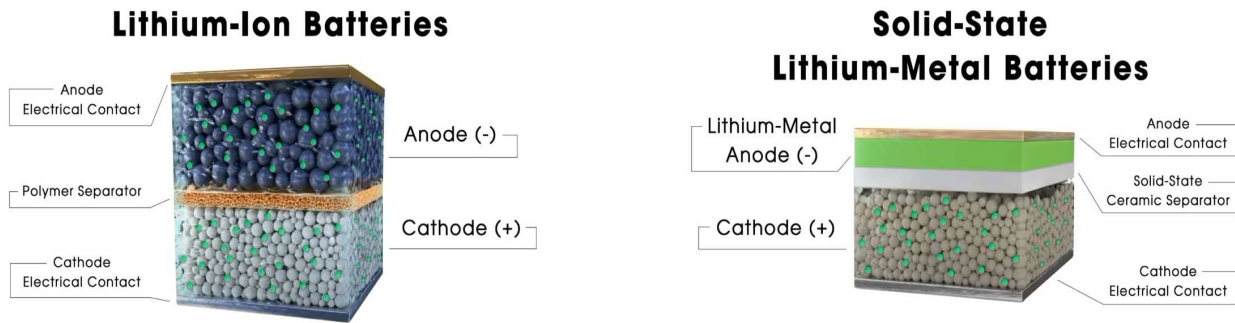
Una vez el mercado empieza a asentarse y los vehículos de combustión ganan la carrera a los eléctricos, en 1950 comienza de nuevo el “boom” de los VEs, en Francia de la mano de Peugeot y en Japón de la mano de Nissan. Pero no será hasta finales de los 70 cuando la crisis del petróleo obliga a la gente a pensar en la dependencia petrolífera que azotaba al planeta. Desde ese momento, la industria de los VEs no ha dejado de crecer, encontrándose con diferentes problemas, rivales (vehículos de hidrógeno) y una proyección a futuro inmensa.

El problema que antes se nombraba como “problema principal” (la autonomía) ahora se ha desarrollado y desglosado en distintos factores: tiempo de carga, velocidades de los cargadores, situación de las EdC, tráfico y espera en las EdC, y el SoC.

Todos los problemas antes mencionados no existirían si las baterías fuesen de una capacidad mayor, siendo del mismo o menor tamaño que las actuales, es decir, que la densidad energética del material del que están hechas las baterías fuera mayor.

Las baterías más usadas en VEs actualmente son las de ion de litio (Li-ion) las cuales tienen una densidad energética de 200 Wh/kg de media. Sin embargo, nuevas tecnologías como las baterías de estado sólido, ya en las primeras fases de prueba, alcanzan los 425

Wh/kg de media, con la visión a futuro de llegar a los 600 Wh/kg.



Nota. Reproducido de "Solid-state batteries: the new frontier of electrification?", por FlashBattery, 2022.

Figura 1.2. Composición de las baterías de ion de litio y estado sólido.

Ya que estas nuevas tecnologías aún están en desarrollo, un buen sistema de EdC resolvería en buena medida el problema de la recarga de los VE. Que sea fácil poder parar con un EV a recargar, sin mucha espera y estando por un corto período de tiempo, son los tres factores que ayudarían al conductor de un EV a despreocuparse por quedarse sin batería en su trayecto. Esta situación presenta dificultades, sobre todo, debido al espacio que ocupan las EdC, en grandes ciudades, donde dicho espacio no abunda, la mayoría de EdC se hallan en los aparcamientos subterráneos y los que no, solo tienen un par de plazas para poder recargar.

La solución que se propone en este trabajo de fin de grado es la mejora de los sistemas de predicción del tráfico y la obtención de un algoritmo que determine dónde sería mejor parar a recargar según el SoC del vehículo en distintos momentos de una ruta determinada, ya que si el sistema optimiza la ruta que se ha de realizar, dirigiendo al conductor del EV al cargador más cercano cuando es necesario, el SoC de dicho EV se maximizará. El cálculo de una serie de rutas, comenzando en el punto de inicio, parando si es necesario en una EdC y continuando hasta el punto final de la ruta, determinará una descarga distinta en cada caso, obteniendo datos del SoC en cada uno de los puntos mencionados. La obtención de estos datos es necesaria para calcular un algoritmo que sea afín a todos los EVs y todas las rutas que pueden escoger los conductores.

Para realizar esto se han escogido dos zonas de la provincia de Málaga, teniendo en cuenta la fluctuación del tráfico, la cantidad de EdC por kilómetro cuadrado, el tipo de vías que hay en la zona (primarias, secundarias, autovías, etc.) y la razón de la zona en sí (residencial, comercial, industrial, etc.).

Se han diseñado dos simulaciones distintas, una por cada zona, utilizando dos herramientas distintas (Python® y AimSun Next® (figura 1.3)) para la obtención más completa de los algoritmos antes mencionados, además de una serie de datos expuestos en los capítulos posteriores.



Figura 1.3. Programas utilizados para la realización de las simulaciones.

El fin de estas simulaciones es tener una idea lo más real posible de la dinámica del tráfico en estas zonas de Málaga y, junto al resto de datos, poder saber dónde deberían localizarse más EdC para abastecer toda la demanda energética. Teniendo en cuenta que esta demanda alteraría la curva de demanda y potencia eléctrica global de todos los ciudadanos, este trabajo de fin de grado trata de optimizar lo máximo posible económica y especialmente dichas localizaciones de cargadores.

Capítulo 2

Metodología y programas

Este capítulo está dedicado a la explicación de la metodología que se ha seguido para realizar las simulaciones, además de la explicación en detalle de los programas utilizados. Antes de comenzar con la explicación en sí de los programas, se menciona de forma breve, que ambas herramientas de simulación eran desconocidas por el redactor de este trabajo de fin de grado antes del comienzo de dicho trabajo. Se toma como algo importante a remarcar, ya que se ha llevado a cabo un aprendizaje paralelo, desde cero, para comprender, manejar y poder utilizar las herramientas que se mencionan posteriormente.

2.1. Python® , base de datos, mapeado y rutas

Python® es “un lenguaje de programación interactivo, interpretado y orientado a objetos” según su propia wiki [2], así es como se describe esta herramienta, aunque se toma con mayor profundidad la siguiente descripción: Python® combina una potente capacidad de compilación con una sintaxis muy fácil de comprender. Tiene módulos, clases, excepciones y un nivel muy alto de tipos de datos dinámicos, además de los diferentes estilos de escritura. La característica de Python® que más interesa en este trabajo de fin de grado es el alto nivel de tipos de datos dinámicos, es crucial ya que se tratan tipos de datos muy diferentes, siendo así diferentes porcentajes del cero al cien en el SoC de los vehículos eléctricos, kilómetros recorridos, coordenadas del sistema UTM, valores booleanos, etc. Gracias al alto nivel de combinación de datos y transformación de los mismos, la simulación se puede llevar a cabo,

gracias también al uso de otra de las grandes características del lenguaje Python® , las librerías. En el entorno de las bases de datos, la importancia de las librerías es esencial: “hay interfaces hacia muchas llamadas a diferentes sistemas y librerías” [2].

2.1.1. Librerías utilizadas

En esta simulación se utiliza una serie de librerías que se separan en dos grupos generales: **el análisis de datos y la representación de los datos sobre un mapa.**

En cuanto al análisis de datos, la primera librería es **Numpy**[3] (Numerical Python) donde se encuentran los tipos de dato “ndarray”. Se utilizan para representar vectores y matrices de la dimensión que se desee. También brinda funciones para leer y escribir datos desde la memoria, además de los primeros algoritmos numéricos básicos.

La siguiente librería es **Pandas**[4] (Panel Data), la cual provee el sistema de almacenamiento básico (clases) donde se diferencian dos tipos de datos: las Series y los DataFrame. Ambos tipos de datos serán tablas donde se indexan datos, los cuales, más adelante se representarán de manera sencilla aplicando otra librería.

Finalmente, dentro del análisis de datos, se encuentra la librería **Matplotlib**[5], la librería por excelencia para la representación gráfica básica de Python® .

Ahora, entrando en las librerías para la representación sobre un mapa dinámico, se encuentran las siguientes: OSMnx y NetworkX.

OSMnx[6] es un paquete de Python® para descargar, modelar, analizar y visualizar fácilmente redes de calles y otras características geoespaciales de OpenStreetMap. Puedes descargar y modelar redes para peatones, conductores o ciclistas con una sola línea de código, y luego analizarlas y visualizarlas. También puedes trabajar fácilmente con servicios urbanos/puntos de interés, trazados de edificios, paradas de transporte público, datos de elevación, orientación de calles, velocidad/tiempo de viaje y rutas [7].

NetworkX[8] es un paquete de Python® para la creación, manipulación y estudio de la estructura, dinámica y funciones de redes complejas [9].

Con estas 5 librerías, junto con alguna menos relevante (files, choice) se completa la base para la realización de la simulación con el lenguaje Python® .

2.2. AimSun Next® , métodos y restricciones

AimSun Next® es un software para realizar soluciones digitales de movilidad. Trata la predicción de la evolución del comportamiento de una red de transporte determinada, ayudando a comprender el flujo de tráfico. Dando como entrada la zona de estudio, un flujo de vehículos de entrada y de salida, una serie de centroides (zonas de inicio y fin de ruta donde se dirigirán los vehículos) y el tipo de tratamiento de datos que se quiere aplicar en la simulación. Con AimSun Next® , se simulará de manera más real la velocidad de los vehículos, la cantidad de vehículos por zona, el flujo de entrada y salida de vehículos en cada sector y la distinción entre tipos de vehículos (coches, motos, camiones, etc.).

2.2.1. Métodos de simulación

Dentro de AimSun Next® se encuentran variantes diferentes de modelado, enmarcando la simulación en: micro, meso, macro estático y demanda de transporte. Estas opciones del tratado de flujo de tráfico determinan el nivel de detalle con el que se realizará la simulación.

Estos tipos de modelado se diferencian en la manera en la que se trata al flujo de vehículos, siendo “micro” un tratado específico e individual de cada vehículo pudiendo controlar la velocidad, la aceleración, el frenado, los cambios de carril, las interacciones con peatones y los semáforos. Este tipo de simulaciones son idóneas cuando se busca procesar datos con densidades específicas en zonas muy concretas de las rutas.

El tipo de simulación “meso” trata los datos agrupando los vehículos en pequeñas unidades, considerando el comportamiento de dichas unidades como dinámico, siendo menos real que el tratado micro, aún se consideran simulaciones bastante afines a la realidad. La diferencia con el tipo micro es la velocidad de procesado y del propio simulado, siendo esta (meso) más veloz.

Finalmente, la simulación de tipo “macro estático” basa la simulación en flujos de tráfico y no en vehículos individuales o grupos de ellos. Este tipo de tratamiento de datos se usa principalmente para grandes cantidades de vehículos y en simulaciones que abarcan un largo período de tiempo.

La demanda de transporte representa la cantidad de viajes que se generan y va incluido

como requisito obligatorio en las tres formas de tratado de datos. Esto incluye las matrices OD (figura 2.1) (origen-destino) donde se determina la cantidad de vehículos por hora desde el punto inicial de una ruta hasta todos los puntos finales posibles en la simulación.

	14172: EII	14173: PUERTO DE LA TORRE	14174: TEATINOS	14176: EL VISO	14177: CAMPUS UMA	14196: DIRECCIÓN CÁDIZ	14204: DIRECCIÓN MÁLAGA	Total
14172: EII		300	500	500	400	700	700	3100
14173: PUERTO DE LA TORRE	150		50	30	150	50	100	530
14174: TEATINOS	500	10		150	400	100	100	1260
14176: EL VISO	15	20	150		350	100	200	835
14177: CAMPUS UMA	50	50	400	50		400	300	1250
14196: DIRECCIÓN CÁDIZ	200	10	50	300	300		200	1060
14204: DIRECCIÓN MÁLAGA	300	20	50	200	50	200		820
Total	1215	410	1200	1230	1650	1550	1600	8855

Figura 2.1. Ejemplo de matriz OD

Una vez definido el tipo de simulación que se quiere realizar, empieza la creación de la zona (entendiendo como zona el terreno donde se va a realizar la simulación), existiendo dos maneras de hacerlo desde cero, es decir sin ninguna base real, o exportando desde el sistema OpenStreetMap (OSM) [10] un sector físico real. Cuando se importa una zona desde OSM, primero hay que acceder a la web para descargar el archivo .osm (figura 2.2) del sector donde se quiere realizar la simulación, una vez se tiene dicho “mapa”, se puede importar a AimSun Next® para comenzar con la simulación.



Figura 2.2. Web de OpenStreetMap y ortofoto de la zona.

2.2.2. Restricciones

Es importante tener en cuenta las restricciones que tienen ambas licencias (OSM y AimSun Next®) para la realización de estas simulaciones, ya que limitan tanto el tamaño de la zona que se puede importar desde OSM, como el número de nodos, tamaño y flujo de las redes de tráfico, tiempo de simulación y algunos factores menos relevantes. Teniendo un límite en OSM de 50.000 nodos y en AimSun Next® un límite de simulación y creación de 100 nodos (se toma nodo como cada una de las uniones o bifurcaciones de mínimo dos carriles).

Capítulo 3

Casos de estudio y resultados

Este capítulo está dedicado al desarrollo y análisis de las simulaciones implementadas en el marco de este Trabajo de Fin de Grado. A lo largo del capítulo, se describen con detalle los diferentes casos de estudio planteados, así como la metodología seguida para su ejecución. En particular, se profundiza en el uso de las herramientas Python® y AimSun Next®, explicando cómo se han utilizado para modelar, simular y evaluar los distintos escenarios propuestos. Se ofrece una visión clara y estructurada del proceso de simulación, desde la configuración inicial hasta la obtención e interpretación de los resultados, con el objetivo de proporcionar una comprensión completa del enfoque adoptado.

3.1. Caso 1: Simulación con Python®

Esta simulación consiste en la creación de 20 rutas de entre 5 y 25 kilómetros, donde 20 VEs de distintas características (peso, potencia motora e identificación) recorren dichas rutas en 2 tramos. Si al final del primer tramo el EV tiene menos de un 97%, el algoritmo reconduce al coche hacia la EdC más cercana. Una vez recargada la batería hasta el 100%, se redirige al destino por el camino más corto posible.

3.1.1. Configuración inicial

La configuración inicial de la simulación se compone de varios elementos: las librerías utilizadas, la lista de vehículos con distintas características, las coordenadas UTM de EdC

reales (se utilizan estratégicamente 10 EdC en puntos por donde la afluencia de rutas es mayor) y la descarga de la red de tráfico en Málaga.

Las **librerías** utilizadas, comentadas en el capítulo 2, se observan en figura 3.1

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
!pip install osmnx geopy
import osmnx as ox
import networkx as nx
from google.colab import files
from random import choice
from geopy.distance import great_circle
```

Figura 3.1. Librerías utilizadas.

En modo aclaratorio, las últimas tres librerías (`files`, `choice` de `random` y `great_circle` de `geopy`) son librerías de apoyo para realizar pruebas: `choice` elige aleatoriamente, entre los valores que se le indiquen al comando, una serie de coordenadas. También, son librerías para aplicar restricciones: `geopy.distance` es una librería para calcular las distancias en metros y el apartado `great_circle` calcula una distancia sobre un perímetro circular, para que ningún punto (de inicio o destino) esté pegado a otro. En el caso de esta simulación, esta restricción es de 500 metros.

Para definir los **VEs** se ha utilizado una lista con 3 tipos de datos que definen a cada VE, la numeración (del 1 al 20), el peso (desde 1100 kg hasta 1800 kg) y la potencia motora (desde 60 kW hasta 200 kW).

La descarga de la **red de tráfico** se realiza con el comando `ox.graph_from_place` de la librería OSMnx. Se necesitan dos datos de entrada para poder ejecutarlo: el nombre del lugar (Málaga, Málaga, Spain) y el tipo de red (`network_type="drive"`).

Finalmente, para el **posicionamiento de las EdC**, se ha utilizado una base de datos de EdC en toda España. Electromaps [11] es un sistema de registro de EdC con más de 500.000 EdC registrados en territorio español, según su propia web “brindan servicios de carga inteligente de vanguardia para conductores de vehículos eléctricos y operadores de puntos de carga por igual, ya sean empresas, municipios o servicios públicos”.

Para el fin de tener una simulación lo más afín a la realidad, se han escogido 10 EdC de esta base de datos, eligiendo las zonas más céntricas y por las cuales pasan más vehículos en las simulaciones iniciales.

Se han realizado una serie de intentos, creando rutas aleatorias, para ver cuales eran las zonas donde la cantidad de rutas era mayor, para determinar las EdC más convenientes (figura 3.2).



Figura 3.2. Estaciones de carga seleccionadas.

3.1.2. Algoritmo de cálculo del SoC

Para el cálculo de este algoritmo se tendrán en cuenta únicamente las características antes introducidas en la lista de VEs, ya que el consumo dependerá de estos factores. En caso de querer añadir factores para una simulación 100 % realista solo se han de añadir los datos elegidos a la lista de características y luego añadirlos al algoritmo. Sería interesante añadir factores como viajes con ventanillas bajas (alterando la resistencia al movimiento), con más

de una persona por EV (alterando el peso), tipo de pavimento y temperatura del mismo e incluso el uso del aire acondicionado.

El algoritmo (figura 3.3) se basa en la adición de consumo, de porcentaje por kilómetro recorrido, a una base de 0.8 %/km según unas restricciones impuestas.

Si el EV pesa más de 1300 kg, este consumirá 0.01 %/km por cada 100 kilos que supere este límite. También, se añade una restricción por potencia motora. Si el EV tiene una potencia de más de 150 kW, el consumo aumentará en 0.005 %/km. Si el EV tiene una potencia inferior a 150 kW y superior a 80 kW, este consumirá 0.0025 %/km más que los EVs que tengan una potencia inferior a 80 kW.

```
class VehiculoElectrico:
    def __init__(self, id, peso, potencia, soc_inicial=100):
        self.id = id # Identificador del vehiculo
        self.peso = peso # kg
        self.potencia = potencia # kW
        self.soc = soc_inicial # porcentaje (100% al inicio)

    def consumo_por_km(self):
        base= 0.8 # consumo base en % por km

        # Ajuste por peso
        if self.peso > 1300:
            exceso_kg = self.peso - 1300
            base += (exceso_kg // 100) * 0.01

        # Ajuste por potencia motora
        if self.potencia > 150:
            base += 0.005

        elif 80 <= self.potencia <= 150:
            base += 0.0025

        return base # en % por km

    def recorrer(self, distancia_km):
        consumo = self.consumo_por_km() * distancia_km
        self.soc -= consumo
        self.soc = max(self.soc, 0)
        return self.soc

    def recargar(self):
        self.soc = 100
```

Figura 3.3. Algoritmo de cálculo del SoC.

3.1.3. Bucle de cálculo de rutas

Para la realización del cálculo de las rutas se ha utilizado un bucle *for*, ya que se conoce el fin de las iteraciones (20 rutas), el uso de este tipo de bucles es lo óptimo en este caso.

Dentro de este bucle se pueden encontrar 3 ramas de cálculo: la primera rama (figura 3.4) donde se observan las transformaciones de las coordenadas (inicio de ruta, destino de ruta y EdCs seleccionadas) en nodos dentro de la red de tráfico. La segunda rama (figura 3.5) donde se observa el cálculo de consumo y kilómetros recorridos del primer tramo de la ruta. Y la tercera rama (figura 3.6) donde se observa el cálculo de kilómetros recorridos en el segundo tramo de la ruta.

```
for i in range(20):
    coord_ini_ruta = coord_ini[i]
    coord_des_ruta = coord_des[i]

    nodo_ini = ox.distance.nearest_nodes(graph, X=coord_ini_ruta[0], Y=coord_ini_ruta[1])
    nodo_des = ox.distance.nearest_nodes(graph, X=coord_des_ruta[0], Y=coord_des_ruta[1])

    charging_station_name, charging_station_coords = choice(list(charging_stations.items()))
    charging_station_node = ox.distance.nearest_nodes(graph, X=charging_station_coords[0],
                                                       Y=charging_station_coords[1])
```

Figura 3.4. Rama de transformación de coordenadas a nodos.

```
# Calcular distancia y consumo del primer tramo
longitud_m1 = sum(graph[u][v][0]['length'] for u, v in zip(ruta_parte1[:-1], ruta_parte1[1:]))
longitud_km1 = longitud_m1 / 1000

vehiculo = vehiculos[i]
soc_antes = vehiculo.soc
peso = vehiculo.peso
potencia = vehiculo.potencia
consumo_km = round(vehiculo.consumo_por_km(), 4)

vehiculo.recorrer(longitud_km1) # Primer tramo: consume batería
soc_despues_tramo1 = vehiculo.soc
```

Figura 3.5. Rama de cálculo del primer tramo de la ruta.

```

# Segundo tramo: desde estación al destino
longitud_m2 = sum(graph[u][v][0]['length'] for u, v in zip(ruta_parte2[:-1], ruta_parte2[1:]))
longitud_km2 = longitud_m2 / 1000
vehiculo.recorrer(longitud_km2)

```

Figura 3.6. Rama de cálculo del segundo tramo de la ruta.

Antes del cálculo de cada tramo se utiliza la función *try* para realizar el cálculo de la ruta en sí, uniendo el nodo de inicio con el punto de carga y el punto de carga con el nodo de fin siempre que sea posible (figura 3.7).

```

try:
    ruta_parte1 = nx.shortest_path(graph, nodo_ini, charging_station_node, weight='length')
    ruta_parte2 = nx.shortest_path(graph, charging_station_node, nodo_des, weight='length')
    ruta_completa = ruta_parte1[:-1] + ruta_parte2

```

Figura 3.7. Unión de tramos y creación de la ruta completa.

3.1.4. Resultados obtenidos con Python®

Una vez aplicado el algoritmo y el bucle que calcula la ruta de cada VE, se procede a la adición al mapa de la red de tráfico de Málaga. Con esto y la aplicación de la librería OSMnx, se obtiene una representación gráfica de las 20 rutas con los puntos de inicio, destino y EdC por donde pasan (figura 3.9).

Dentro del bucle del punto 3.1.3. también se realiza un seguimiento de cada ruta, donde se obtiene como resultado una serie de frases exploratorias de los movimientos de cada vehículo. Se puede observar un ejemplo de esto en figura 3.8.

```
[Vehículo 3] Recargando al llegar a E7 (SOC < 97%)
```

Figura 3.8. Ejemplo de mensaje impreso por pantalla de cada ruta.

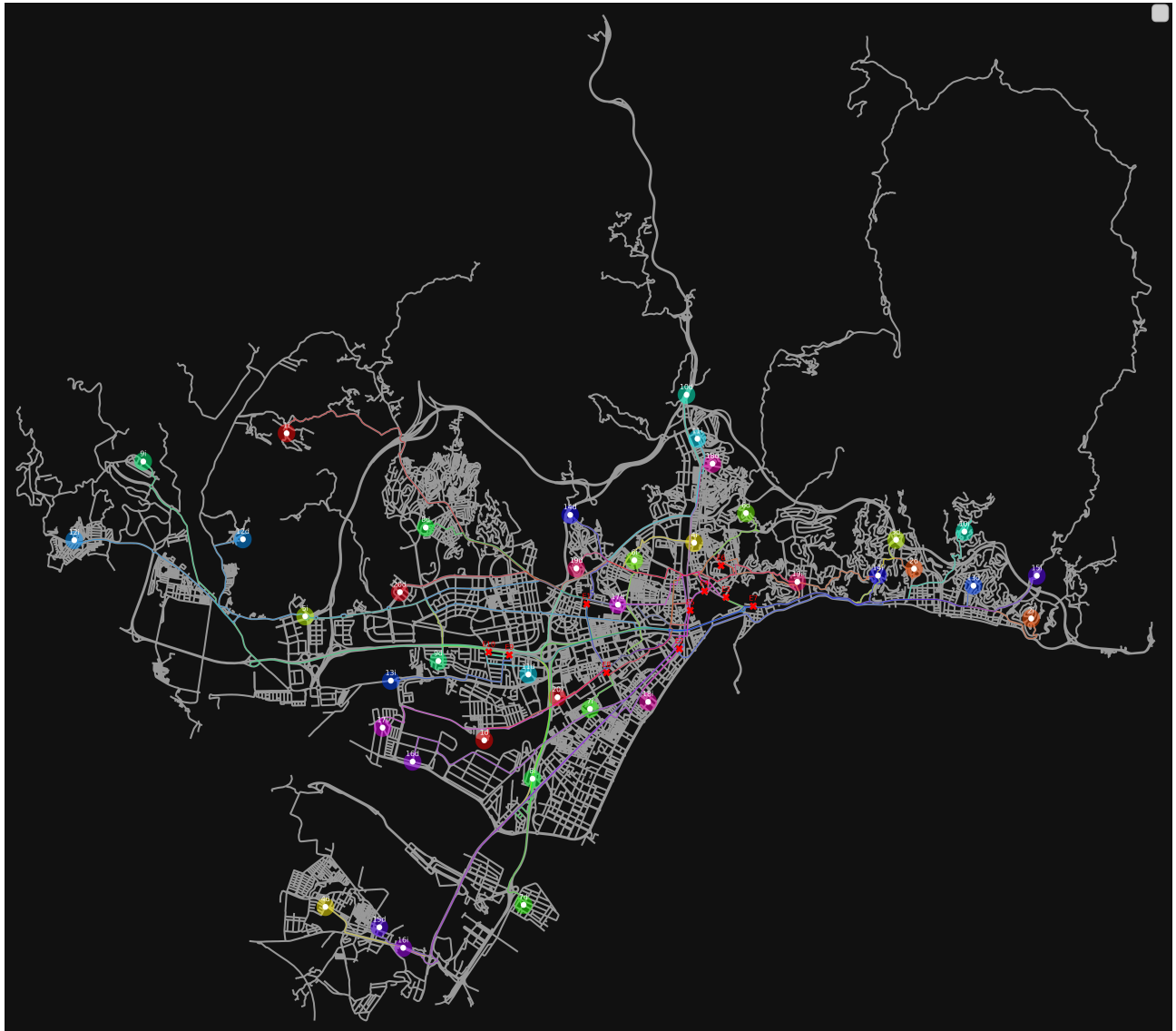


Figura 3.9. Mapa con las 20 rutas.

A partir de este conjunto de rutas, paradas y puntos de inicio y fin, se pueden obtener distintas tablas y gráficas con resultados muy interesantes para el análisis de datos en esta simulación.

En figura 3.10 se muestran recogidos en una gráfica de barras la distancia recorrida por cada uno de los EVs, en el eje de abscisas se encuentra la numeración de cada uno de los EVs y en el eje de ordenadas se encuentra la escala en kilómetros.

En figura 3.11 se muestran tres momentos del SoC de cada uno de los VE, la parte marcada en azul sería el SoC inicial (el estado inicial es del 100%), la parte marcada en verde sería el

estado del SoC de cada EV después de recorrer el tramo 1 de su ruta y la parte marcada en naranja es el estado del SoC al final de cada ruta.

En figura 3.12 se muestran recogidos por cantidad de VE, el SoC final en intervalos de un 1 %, es decir, los EVs que tienen un SoC final de la batería entre '94 %' y '95 %' se encuentran acumulados como un valor en dicha barra. Se muestra una curva de tendencia de los SoC de todas las baterías y una recta en rojo que muestra el umbral impuesto de recarga (porcentaje de batería que si es inferior a 97 %, el algoritmo diseña la ruta para poder recargar en la EdC más cercana).

En la tabla 1 se observan los resultados obtenidos con esta simulación, desde los porcentajes parciales del SoC en cada vehículo, el consumo individual calculado con el algoritmo de SoC, si el EV ha parado a recargar y los kilómetros recorridos.

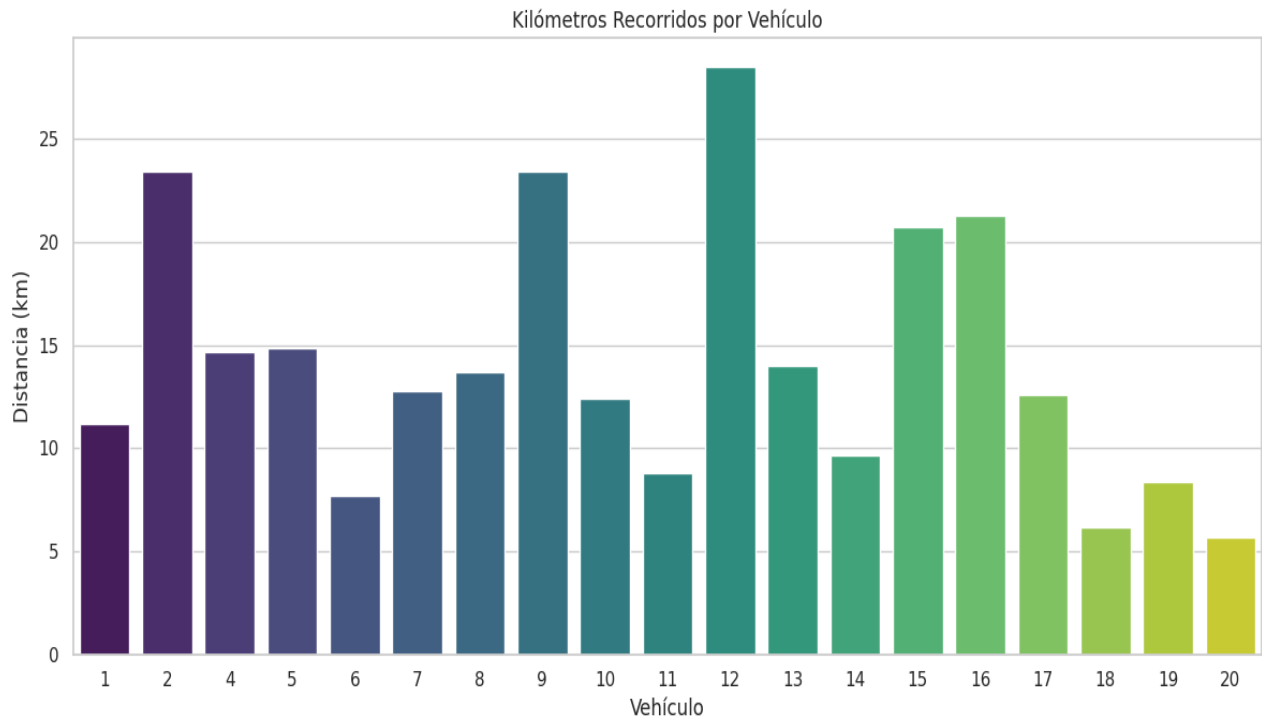


Figura 3.10. Kilómetros recorridos según el vehículo.

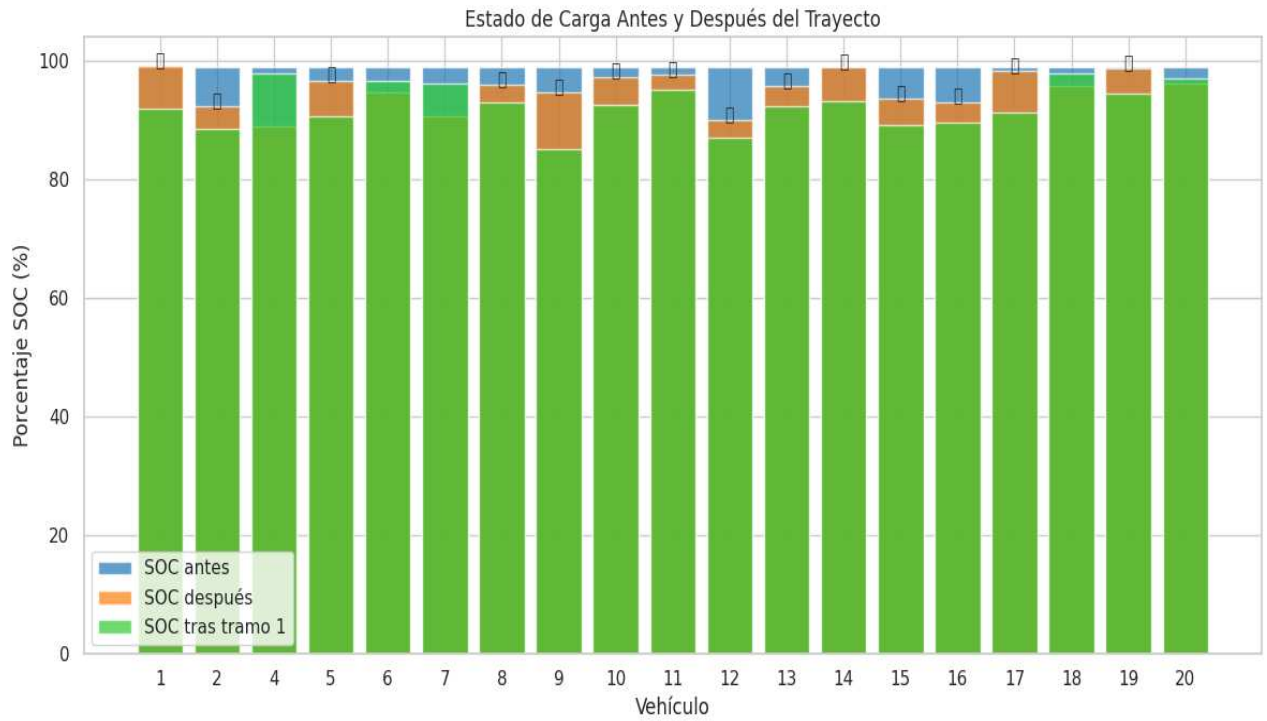


Figura 3.11. Estado del SoC en momentos parciales de la ruta.

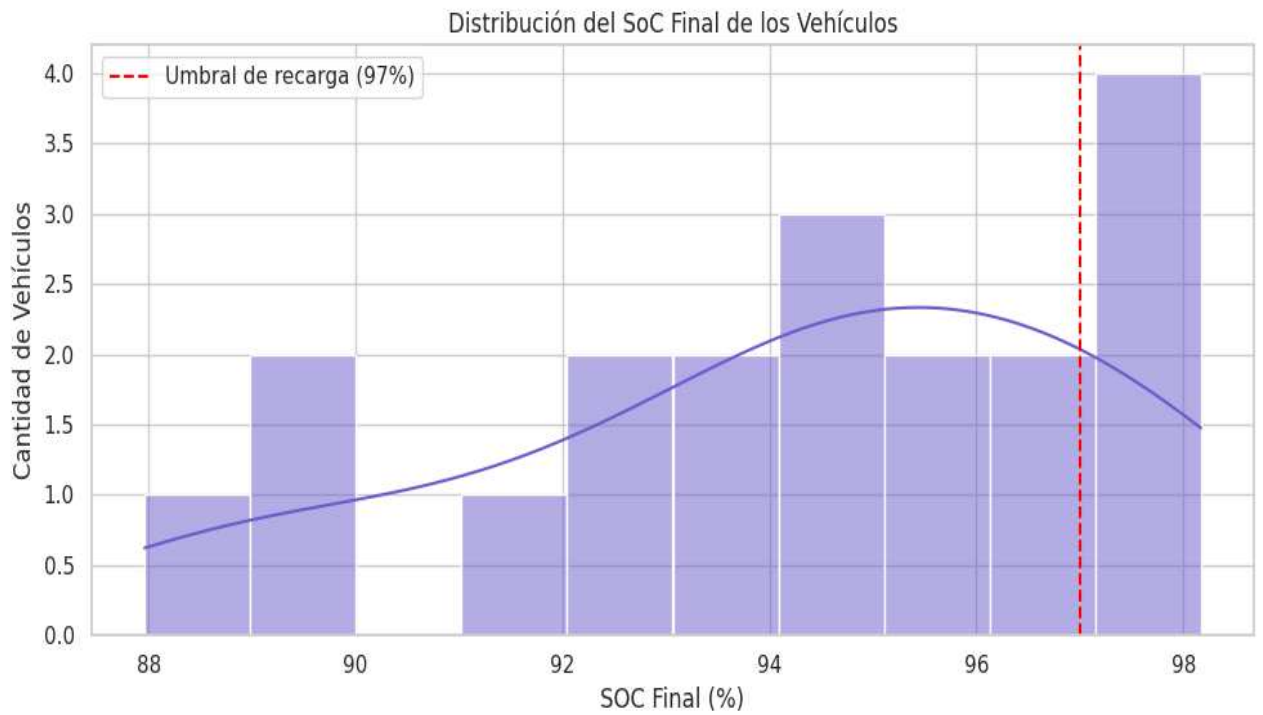


Figura 3.12. Resultados del algoritmo según el SoC final.

Vehículo	Peso (kg)	Potencia (kW)	Consumo por km (%)	SOC Antes (%)	SOC tras tramo 1 (%)	¿Recargó?	SOC Final (%)	Km Recorridos
1	1200	75	0.8	100	92.9	Sí	98.17	11.16
2	1400	100	0.8125	100	89.54	Sí	91.43	23.42
4	1500	130	0.8225	100	98.98	No	87.96	14.64
5	1700	180	0.845	100	91.76	Sí	95.71	14.83
6	1250	85	0.8025	100	97.63	No	93.83	7.69
7	1100	70	0.8	100	97.19	No	89.76	12.8
8	1350	120	0.8025	100	94.03	Sí	94.96	13.71
9	1800	200	0.855	100	86.2	Sí	93.79	23.4
10	1450	140	0.8125	100	93.53	Sí	96.37	12.43
11	1550	110	0.8225	100	96.12	Sí	96.63	8.81
12	1300	95	0.8025	100	88.11	Sí	89.05	28.46
13	1750	170	0.845	100	93.45	Sí	94.72	14
14	1150	60	0.8	100	94.29	Sí	97.97	9.67
15	1650	150	0.8325	100	90.1	Sí	92.66	20.72
16	1400	145	0.8125	100	90.61	Sí	92.1	21.28
17	1500	155	0.825	100	92.3	Sí	97.3	12.61
18	1600	135	0.8325	100	98.83	No	94.83	6.21
19	1250	80	0.8025	100	95.56	Sí	97.74	8.35
20	1800	190	0.855	100	98.09	No	95.14	5.69

Tabla 1. Resumen de vehículos y consumo

Con estos datos concluye la simulación con Python®, teniendo en cuenta los distintos tipos de resultados obtenidos, el análisis de datos que se puede realizar es muy extenso y se puede extender aún más ya que solo se dispone de un nivel bajo de datos reales (es decir una base de datos real de EVs en Málaga) para poder entrenar el algoritmo y mejorarlo hasta el punto que se desee.

3.2. Caso 2: Simulación con AimSun Next®

Esta simulación consiste en la creación de una red de tráfico completa, desde la recreación de las vías hasta la aplicación de flujos de vehículos. Se llevará a cabo con un flujo fijo de vehículos por hora, en un área de entre 82 y 93 km² en la zona oeste de Málaga. La simulación será en el espacio temporal de entre las 08:00 y las 09:00 (una hora de duración). Una vez finalizada la simulación, y siempre que no existan errores de compilación, se obtendrán una serie de datos relevantes (como el tiempo medio de cola, densidad de vehículos por kilómetro, distancia total de viaje, flujo de vehículos por hora, velocidad, etc.) que permitirán realizar el análisis del posicionamiento de las EdC.

Al final, el problema planteado en el capítulo 1 solo puede resolverse abasteciendo con EdC aquellas zonas que presentan mayor flujo y densidad de tráfico.

El objetivo de esta simulación es generar una base de datos que complemente la simulación descrita en el apartado 3.1.. Esto permitirá identificar las áreas con mayor flujo y densidad de tráfico, para dotarlas de EdC y así cubrir la demanda de los EVs que las requieran. Gracias a la simulación realizada con Python®, es posible conocer tanto el momento en que un EV necesita recargar como la elección de la EdC más óptima para cada trayecto.

3.2.1. Base de la simulación

Para comenzar esta simulación, se realiza un análisis de distintas zonas de Málaga, con el objetivo de elegir la zona más completa para tomar como base en la simulación (se tiene en cuenta la restricción vista en el capítulo 2, que limita este tipo de simulaciones a un total de 100 nodos).

Las condiciones impuestas para las opciones son:

1. Salidas y entradas a autovía o autopista.
2. Zona residencial (además de otras).
3. Flujos altos, fijos y constantes a determinadas horas.
4. Tener conocimiento propio de la zona.

La elección de las tres primeras opciones tiene una razón sencilla: la zona debe incluir el mayor número posible de escenarios dentro del tráfico vehicular para abarcar la mayoría de los casos. La última condición responde al mismo motivo, ya que cuanto mejor se conozca la zona seleccionada, más precisa será la aproximación a la realidad.

Finalmente, se seleccionan 4 zonas para hacer la designación de la que mejor cumple dichos requisitos.

3.2.1.1. Zona 1: Málaga centro

Málaga centro se ha tomado como una opción válida, a pesar de no disponer de entradas y salidas directas a autovías o autopistas. Sin embargo, presenta uno de los mayores flujos de vehículos de la ciudad, con un tráfico constante y muy estable. Se trata de una zona residencial ampliamente conocida, que además cuenta con el puerto y varias estaciones de autobús, lo que la convierte en un área de gran relevancia dentro del sistema de movilidad urbana (figura 3.13).



Figura 3.13. Ortofoto de la zona de Málaga centro.

3.2.1.2. Zona 2: Ciudad Jardín norte

Ciudad Jardín (zona norte) ha sido considerada una opción especialmente interesante, ya que combina el carácter residencial con accesos directos a la autovía A-45, lo que incrementa el dinamismo del tráfico en determinados puntos. Esta conexión con la red de alta capacidad genera escenarios variados que incluyen tanto tráfico local como de paso. Su configuración urbana permiten una aproximación realista en la simulación, abarcando una amplia gama de situaciones dentro del comportamiento del flujo vehicular. A pesar de cumplir con todos los criterios, no se tiene apenas conocimiento del flujo real de esta zona (de manera personal). (figura 3.14).



Figura 3.14. Ortofoto de la zona de Ciudad Jardín norte.

3.2.1.3. Zona 3: Puerta Blanca

Puerta Blanca, por su parte, destaca por su proximidad a la MA-21 y su conexión directa con la autovía A-7, lo que incrementa notablemente el flujo de vehículos, especialmente en horas punta. Esta zona combina áreas residenciales, centros educativos y accesos a instalaciones deportivas y comerciales, generando una gran variedad de escenarios de tráfico. El conocimiento previo de la zona favorece una simulación más precisa y representativa de la realidad que otras zonas. (figura 3.15).



Figura 3.15. Ortofoto de la zona de Puerta Blanca.

3.2.1.4. Zona 4: El Viso/Teatinos

La zona de El Viso/Teatinos también se consideró como una opción muy válida debido a su ubicación estratégica y características mixtas. Es una zona con un tráfico elevado en la mayoría de franjas horarias como otras áreas principales de Málaga, su cercanía a la autovía A-7 y su función como vía de conexión entre distintas zonas residenciales le otorgan un flujo vehicular alto y constante. Al ser una zona muy bien conocida y de fácil delimitación, permite una representación adecuada de escenarios de tráfico habituales, lo que contribuye a una simulación muy ajustada a la realidad. (figura 3.16).



Figura 3.16. Ortofoto de la zona de El Viso/Teatinos.

Esta zona se acabó eligiendo como base de la simulación, teniendo en cuenta que el conocimiento personal de la zona es el más elevado de las cuatro zonas, la característica de zona mixta (urbana, residencial y con parques comerciales) es la más completa, su cercanía a la autovía A-7 (entradas y salidas disponibles) y su flujo de vehículos de gran tamaño, estable y fijo, hacen que sea la zona que cumple mejor con las condiciones impuestas.

3.2.2. Adaptación e implantación de la zona

Una vez elegida la zona que se va a simular, se exporta desde la web de OpenStreetMaps [10] la región delimitada escogida, como se observa en figura 3.17, además se observa las opciones de configuración de la simulación para tratar los datos (en este caso se escoge la opción micro como se explicó en el capítulo 2), el sentido de circulación de la zona seleccionada (España, se circula por la derecha) y un apartado desplegable donde se puede importar un mapa directo desde Internet, además de otras opciones (usar un archivo .osm esta dentro de estas opciones).

La razón por la cual se descarga primero el archivo .osm de la web y no se utiliza la opción de “importar mapa desde Internet”, es por la restricción de nodos que limita esta licencia, con lo cual se ha preferido realizar a mano cada una de las carreteras y objetos de tráfico (rotondas, pasos de peatones, limitadores de velocidad, ceda el paso, señales de Stop, etc.) para delimitar, controlar y poder explotar al máximo estas restricciones.

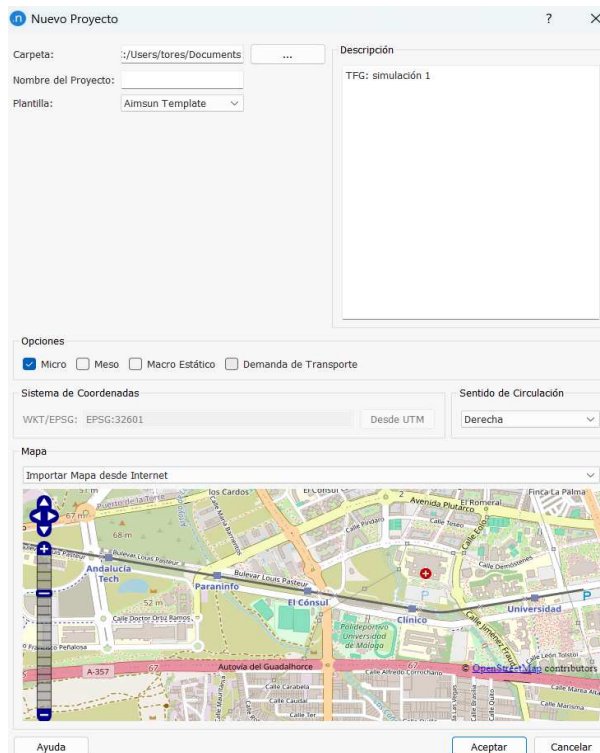


Figura 3.17. Implementación de la zona escogida en AimSun Next® .

3.2.3. Recreación de las vías y objetos de tráfico

Este apartado está dedicado a la explicación de cómo se ha recreado cada una de las vías (todos los tipos) y todos los objetos de tráfico que hay en la zona seleccionada, a partir de la ortofoto, representada en figura 3.16, como plantilla.

3.2.3.1. Vías

Para la recreación de las vías y sus distintos tipos de modificaciones (ver figura 3.18), se comienza utilizando la herramienta de “creación de una sección” para añadir las secciones (vías) que se deseen. Una vez se hace clic en dicha herramienta, se selecciona el punto de inicio de la sección y se procede a hacer clic en cada fin e inicio de nueva sección (creando así una vía). Cuando la vía ya está creada, con el clic derecho se selecciona la vía y se modifica de la manera que se desee, pudiendo cambiar el número de carriles ($\text{ctrl}+\text{n}^\circ$), ya que de manera predeterminada los carriles son dobles, o accediendo al menú de configuración (ver parte derecha de figura 3.18) donde las opciones más usadas en esta simulación son: la variación de la velocidad (por la derecha en algunas zonas a 30 km/h, o 50 km/h en las vías principales), el tipo de vía (primaria, secundaria, vía de servicio, etc.) y la altura de la vía (túneles o secciones por encima del nivel normal del resto).



Figura 3.18. Esquema de creación de vías y sus posibles modificaciones.

Como ejemplo, en la zona marcada con un recuadro rojo de figura 3.19, se recrea un túnel a 5 metros de altura por debajo del nivel del resto de vías de esa micro-zona. Las vías del recuadro de la izquierda en figura 3.19 son ambas vías con la altura modificada.

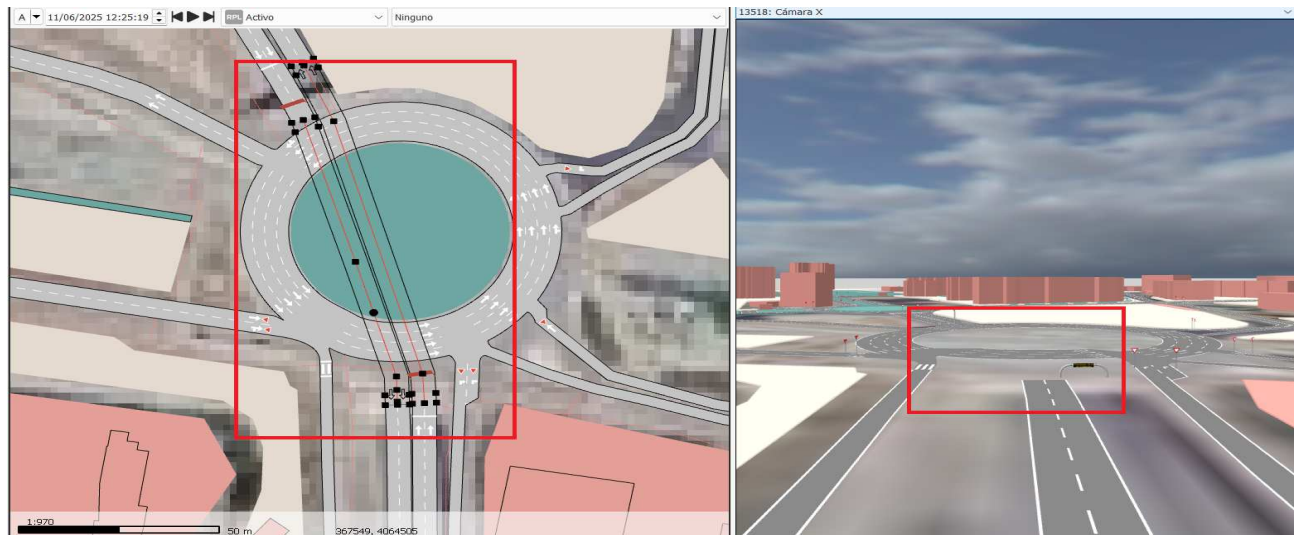


Figura 3.19. Túnel a 5 metros de diferencia con el nivel del resto de las vías.

3.2.3.2. Nodos y rotondas

A la hora de unir dos o más secciones se creará un nodo, con lo cual (como se observa en figura 3.20) cada rotonda tendrá un número de nodos que dependerá de las entradas y salidas de la misma. Cada unificación o bifurcación de carriles formará un nuevo nodo que unirá o separará dos secciones, o más, distintas.

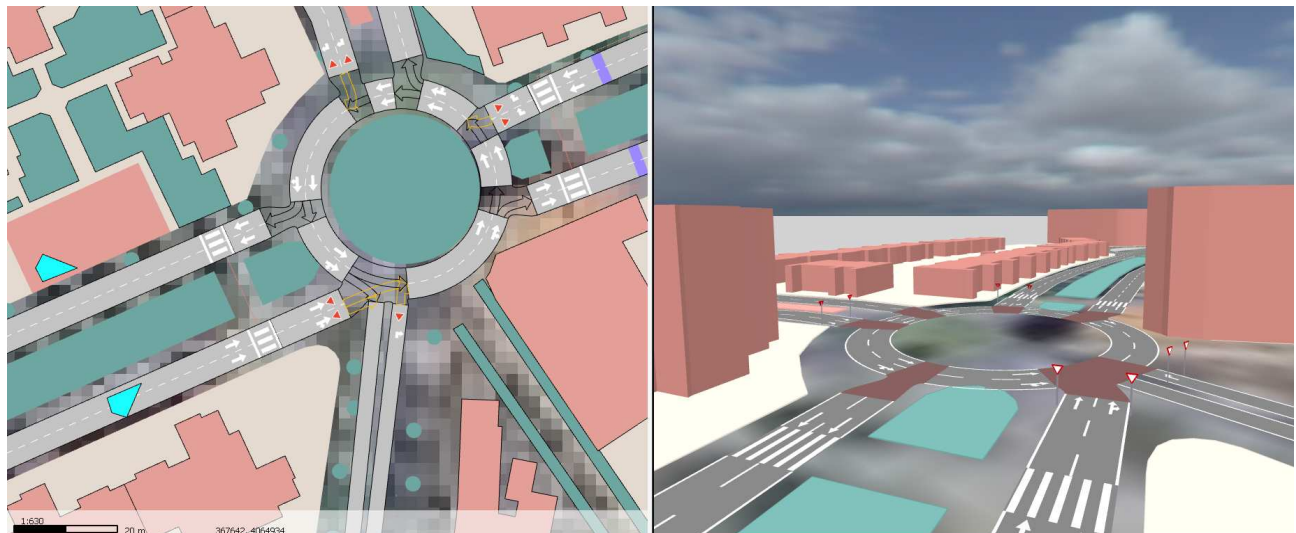


Figura 3.20. Ronda con los nodos seleccionados.

Para la recreación de las rotondas, se utiliza la herramienta “crear una Rotonda”, se seleccionan todos los carriles de entrada y salida de dicha rotonda, luego se hace clic en la herramienta y se posiciona el cursor donde se desee poner el centro de la rotonda, una vez realizado esto, se mueve el cursor para expandir la rotonda hasta que tenga el diámetro que busque. Antes de soltar el clic se presiona “alt+n^o” para seleccionar el número de carriles de la rotonda (se observa en figura 3.21).



Figura 3.21. Modo de implementación de una rotonda.

3.2.3.3. Señales de tráfico

Este apartado es importante en la simulación ya que dará mucha afinidad a la realidad en los tiempos de todas las rutas, teniendo en cuenta que se han recreado todas las señales de ceda el paso y señales de stop que hay en la realidad, además de los pasos de peatones y algunos carteles de advertencia en las zonas de máxima densidad de vehículos. En esta zona no se encuentra ningún semáforo, pero sería igual implementarlo en caso que hubiese. Para añadir cualquier señalización de estos tipos, solo hay que hacer clic en su respectiva herramienta, en la barra de herramientas y seleccionar el lugar donde se desea posicionar en el mapa (ver figura 3.22).

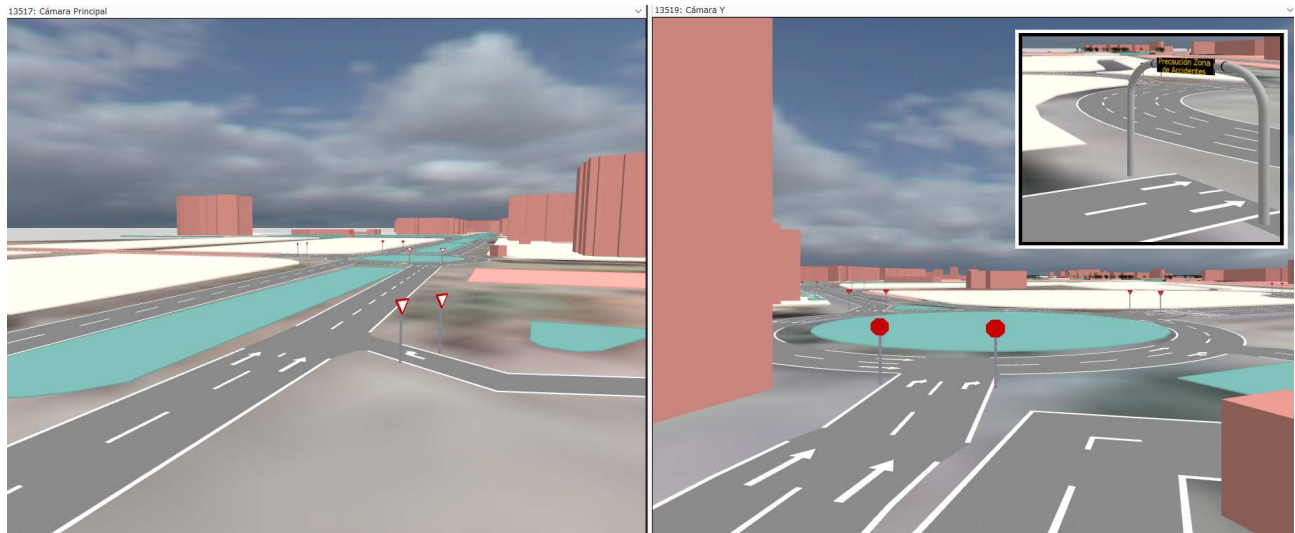


Figura 3.22. Distintas señales recreadas de la realidad.

3.2.4. Cuerpo de la simulación

Este apartado está dedicado a la explicación del cuerpo de la simulación, es decir, la creación de los lugares de destino y origen de los vehículos (centroides) y la manipulación del volumen de desplazamientos entre estos puntos (matriz OD), además de detalles como el porcentaje de elección en cada nodo (giros).

3.2.4.1. Centroides

En AimSun Next® , los centroides [12] son puntos especiales que representan los orígenes y destinos del tráfico dentro de una red de simulación. No son lugares físicos como calles o intersecciones, sino nodos abstractos que indican de dónde vienen o hacia dónde van los vehículos.

Los centroides son como las “puertas” por donde los vehículos entran o salen del modelo de tráfico. Cada zona del modelo (por ejemplo, un vecindario o un área industrial) tiene al menos uno de estos puntos, y se conectan al resto de la red a través de enlaces específicos llamados conectores.

Estos elementos son fundamentales para poder asignar los viajes según una matriz de demanda, es decir, para saber cuántos vehículos se mueven de un lugar a otro en un cierto período de tiempo.

En esta simulación se han implementado un total de siete centroides (se observan en figura 3.23 de izquierda a derecha y de arriba hacia abajo):

1. EII (Escuela de ingenierías industriales)
2. Dirección Cádiz (A-357 para unirse a A-7 con dirección a Cádiz)
3. El Viso (Parque industrial de El Viso)
4. Dirección Málaga (A-357 con dirección a Málaga centro)
5. Campus UMA (Campus de la universidad de Málaga)
6. Teatinos (Barrio de Teatinos)
7. Puerto de la Torre (Barrio de Puerto de la Torre)

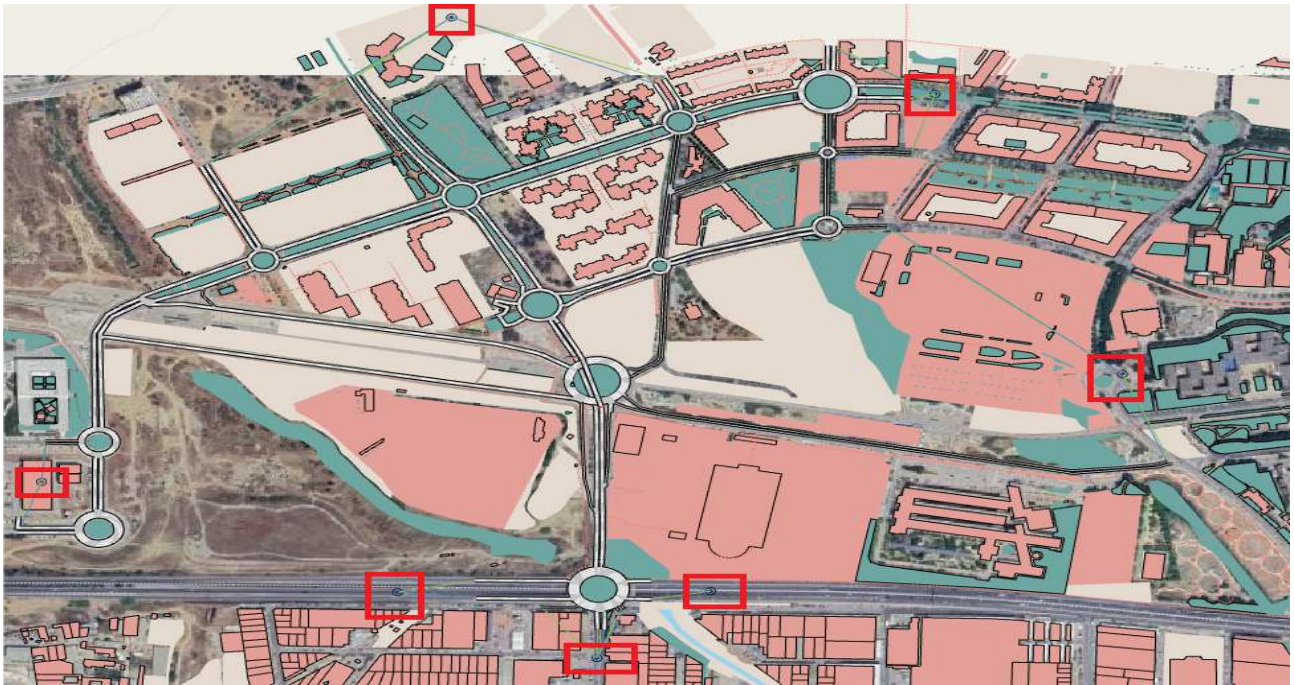


Figura 3.23. Centroides implementados en la simulación.

3.2.4.2. Matriz Origen-Destino

En AimSun Next® (y en modelado de tráfico en general), la matriz Origen-Destino (OD) [12] es una tabla que indica cuántos viajes se realizan desde una zona específica de origen hasta una zona específica de destino durante un periodo determinado.

En figura 3.24 cada fila representa una zona de origen: es decir, el lugar desde donde salen los vehículos. Cada columna representa una zona de destino: es decir, el lugar hacia donde se dirigen los vehículos. Por ejemplo: en la intersección entre la fila “14172: EII” y la columna “14196: DIRECCIÓN CÁDIZ”, vemos el número 700. Esto significa que 700 vehículos salen de la zona EII y se dirigen hacia Dirección Cádiz. El total al final de cada fila (última columna) indica cuántos viajes parten desde esa zona en total. El total al final de cada columna (última fila) indica cuántos viajes llegan a esa zona desde todas las demás.

Esta matriz es esencial porque sirve como punto de partida para simular el comportamiento del tráfico. Al no representar vehículos individuales, sino volúmenes agregados de viajes. Es útil para saber:

1. Cuáles zonas están generando más tráfico.
2. Qué zonas son más atractivas como destino.
3. Cómo se distribuye la movilidad urbana dentro del modelo.

En figura 3.24 se muestra el flujo total de vehículos implementados en esta simulación (coches, motocicletas y camiones) que luego se distribuyen en 3 matrices similares a esta, de las cuales la suma es la matriz que se observa. La proporción impuesta es de 70% coches, 20% motocicletas y 10% camiones. Una vez que se comienza a simular, dependerá del propio flujo de tráfico, el porcentaje final de cada tipo de vehículo, ya que se crearán colas de espera para entrar en la simulación por los centroides de origen donde la densidad del tráfico sea mayor.

Como se mencionó previamente, la matriz OD permite conocer la cantidad de viajes que se realizan desde cada centroide de origen hacia cada centroide de destino dentro de un período específico.

Para esta simulación, se ha seleccionado el intervalo de 08:00 a 09:00 horas, ya que corresponde al momento del día en el que se dispone de mayor información sobre el comportamiento del tráfico en la zona modelada. Esta elección se basa tanto en datos disponibles como en observaciones personales al circular por el área.

	14172: EII	14173: PUERTO DE LA TORRE	14174: TEATINOS	14176: EL VISO	14177: CAMPUS UMA	14196: DIRECCIÓN CÁDIZ	14204: DIRECCIÓN MÁLAGA	Total
14172: EII	160			175	160	205	225	1150
14173: PUERTO DE LA TORRE	95		50	30	100	50	85	410
14174: TEATINOS	325	10		80	210	90	85	800
14176: EL VISO	15	20	135		190	90	75	525
14177: CAMPUS UMA	50	50	340	50		200	145	835
14196: DIRECCIÓN CÁDIZ	170	10	50	150	145		95	620
14204: DIRECCIÓN MÁLAGA	270	20	50	100	50	100		590
Total	325	270	830	585	855	735	710	4830

Figura 3.24. Matriz OD Total de todos los vehículos.

3.2.4.3. Porcentaje de giro

Para diseñar el flujo de tráfico únicamente faltaría la implementación del comportamiento de los conductores, algo muy difícil de plasmar en una simulación de manera realista, ya que cada conductor escoge variantes de su ruta de manera casi aleatoria cuando va hacia su destino (en el caso de elección de giros, salidas de rotondas, alguna vía cortada, accidentes, etc.) Con lo cual se añade un sistema de porcentajes de giro y elecciones de ruta según la estadística en cada nodo (obtenidos de observaciones personales al circular). Este método para acercarse a la realidad en la implementación del comportamiento se observa en figura 3.25, donde en cada nodo se aplica un porcentaje según la estadística de que elige cada uno de los conductores. En el nodo seleccionado en dicha figura, el 70 % de los conductores que van por esa vía toman esa salida de la rotonda y el otro 30 % sigue en la rotonda.

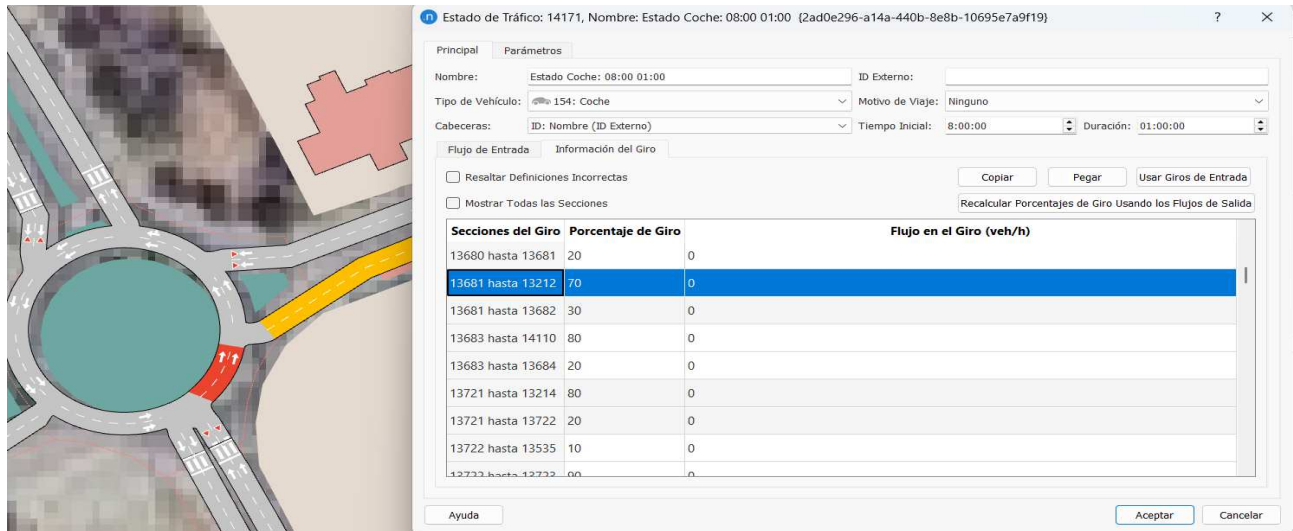


Figura 3.25. Implementación de porcentajes.

3.3. Resultados de la simulación

Una vez finalizados los puntos 3.1. y 3.2. se puede comenzar a simular, añadiendo las matrices OD como demanda de tráfico para juntar todos los tipos de vehículos (figura 3.26). Ahora solo se ha de ejecutar la simulación, introduciendo un escenario dinámico sobre el mapa recreado y aplicando la demanda de tráfico.



Figura 3.26. Inicialización de la demanda de tráfico en la simulación.

Los resultados generados por AimSun Next® se han organizado en tablas para facilitar su análisis e interpretación. Estas tablas permiten comparar el desempeño del sistema bajo las condiciones simuladas, identificar posibles puntos críticos de congestión y evaluar el efecto de diversas estrategias de gestión del posicionamiento de las EdC.

A continuación, se presentan los datos más relevantes obtenidos en la simulación, desglosados por intervalos temporales secciones de la red, y parámetros operacionales.

En la leyenda de cada gráfica se observa el tipo de vehículo en especial y la simulación la cual se está ejecutando.

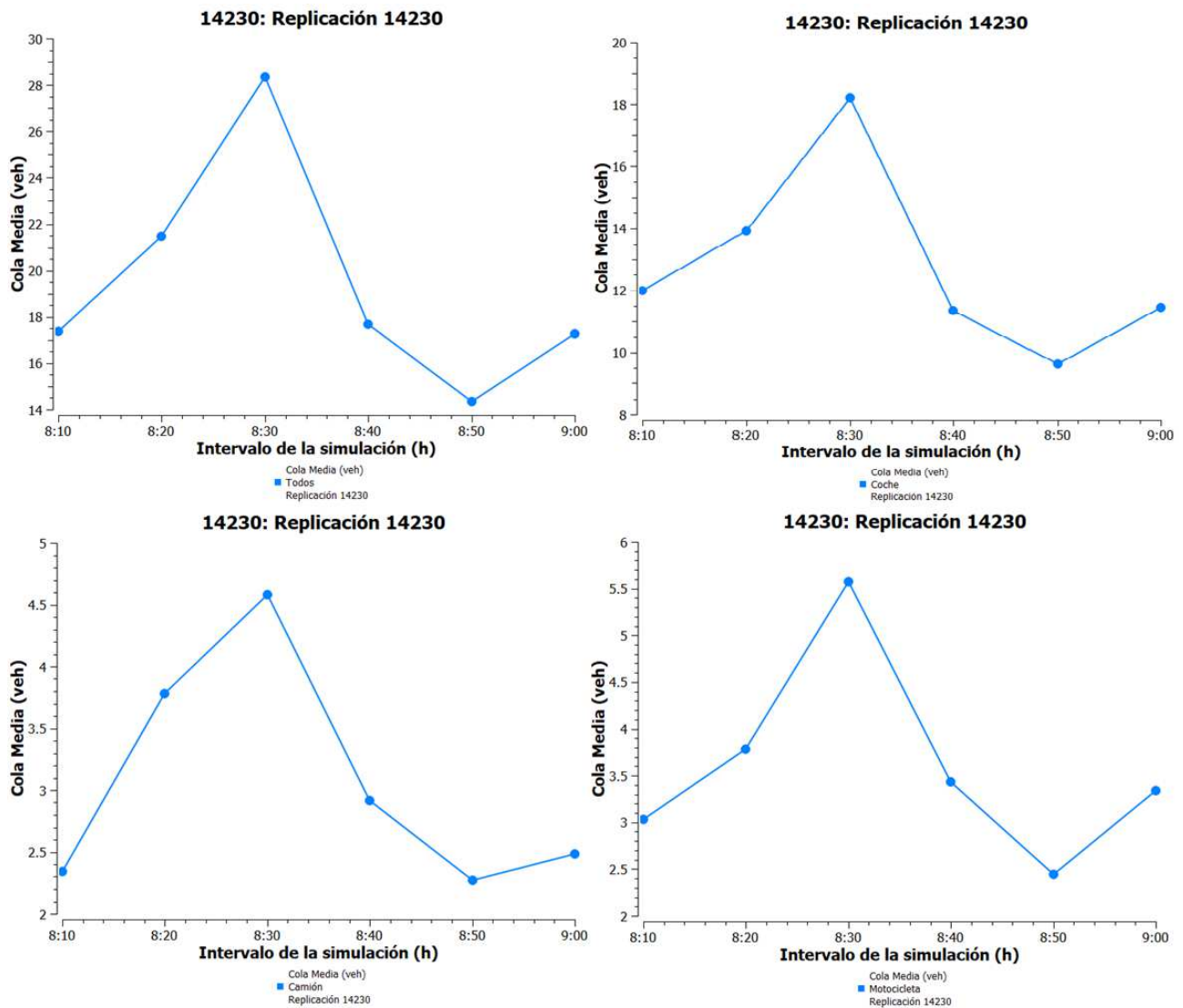


Figura 3.27. Cola media de vehículos en las vías durante el intervalo de la simulación.

En cuanto a la cola media de vehículos (figura 3.27), se observa que a las 08:30 horas es cuando todos los vehículos tienen su máximo punto de cola y de espera en las vías. Sería un momento justo para parar a recargar si un EV lo necesitara, ya que utilizaría ese tiempo entre las 08:20 y las 08:40 (20 minutos) para recargar las baterías y perdiendo menos tiempo que si realiza la recarga en un momento cuando la cola esta en los puntos valle.

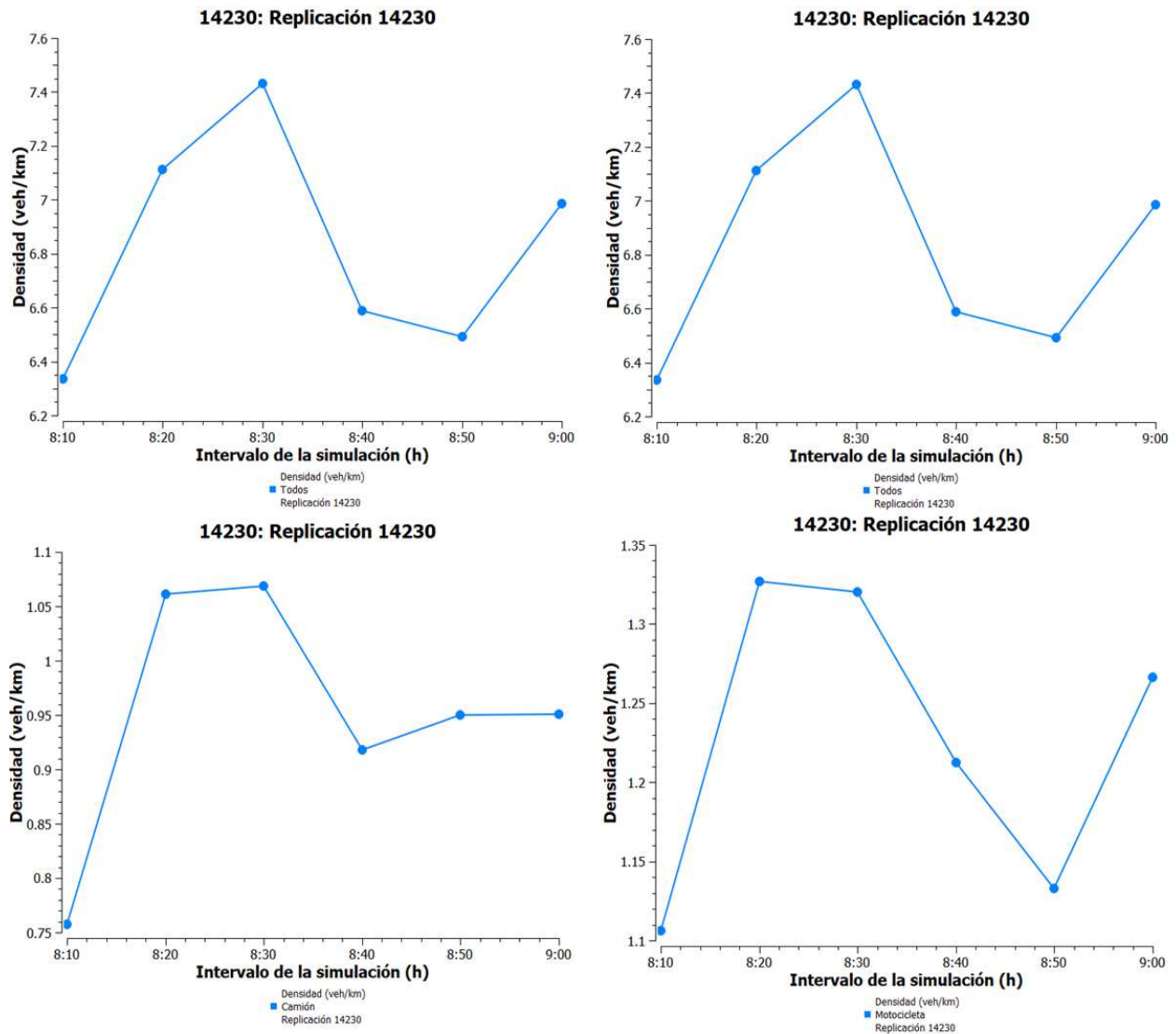


Figura 3.28. Densidad de vehículos en las vías durante el intervalo de la simulación.

En los parámetros de densidad se observa en figura 3.28 un claro repunte desde las 08:20 hasta la 08:40, esto esta directamente relacionado con el flujo de vehículos (figura 3.29), lo cual ocasiona las colas en las vías en torno a las 08:30.

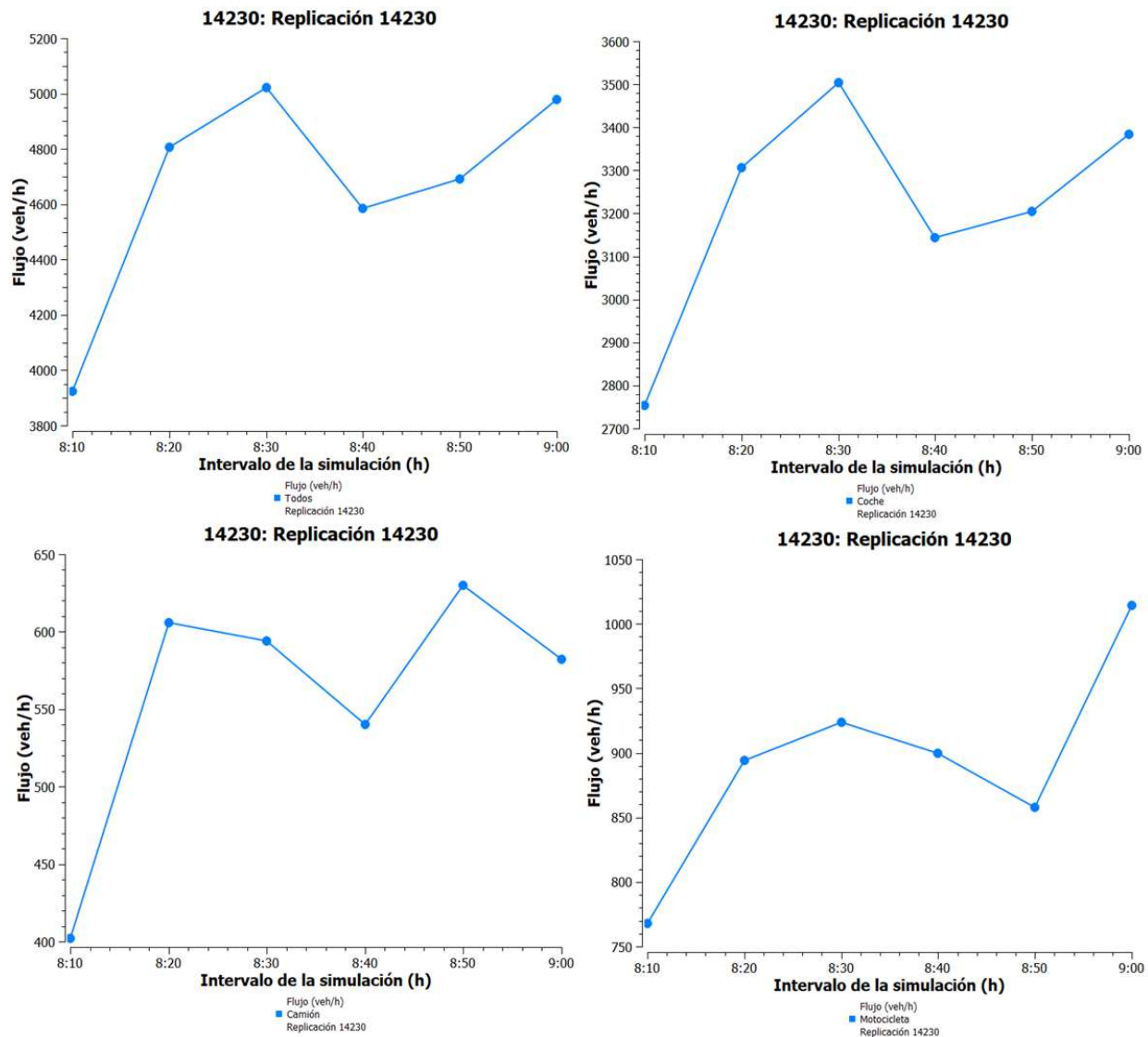


Figura 3.29. Densidad de vehículos en las vías durante el intervalo de la simulación.

El flujo de vehículos (figura 3.29) presenta una leve bajada debido a la espera de los vehículos que no han accedido aún a la zona simulada, por eso el total de vehículos son 4930 (figura 3.24) y no 8855 (figura 2.1) que sería el total de la simulación si no estuviese enmarcada en un intervalo de tiempo.

Complementando a la figura 3.29 se observa una vista cenital de distintos puntos de la simulación con un resalto en distintos colores (figura 3.30). Resaltando en tonos más rojizos las zona con mayor flujo de vehículos y las zonas verdes las zonas con menor flujo de vehículos.

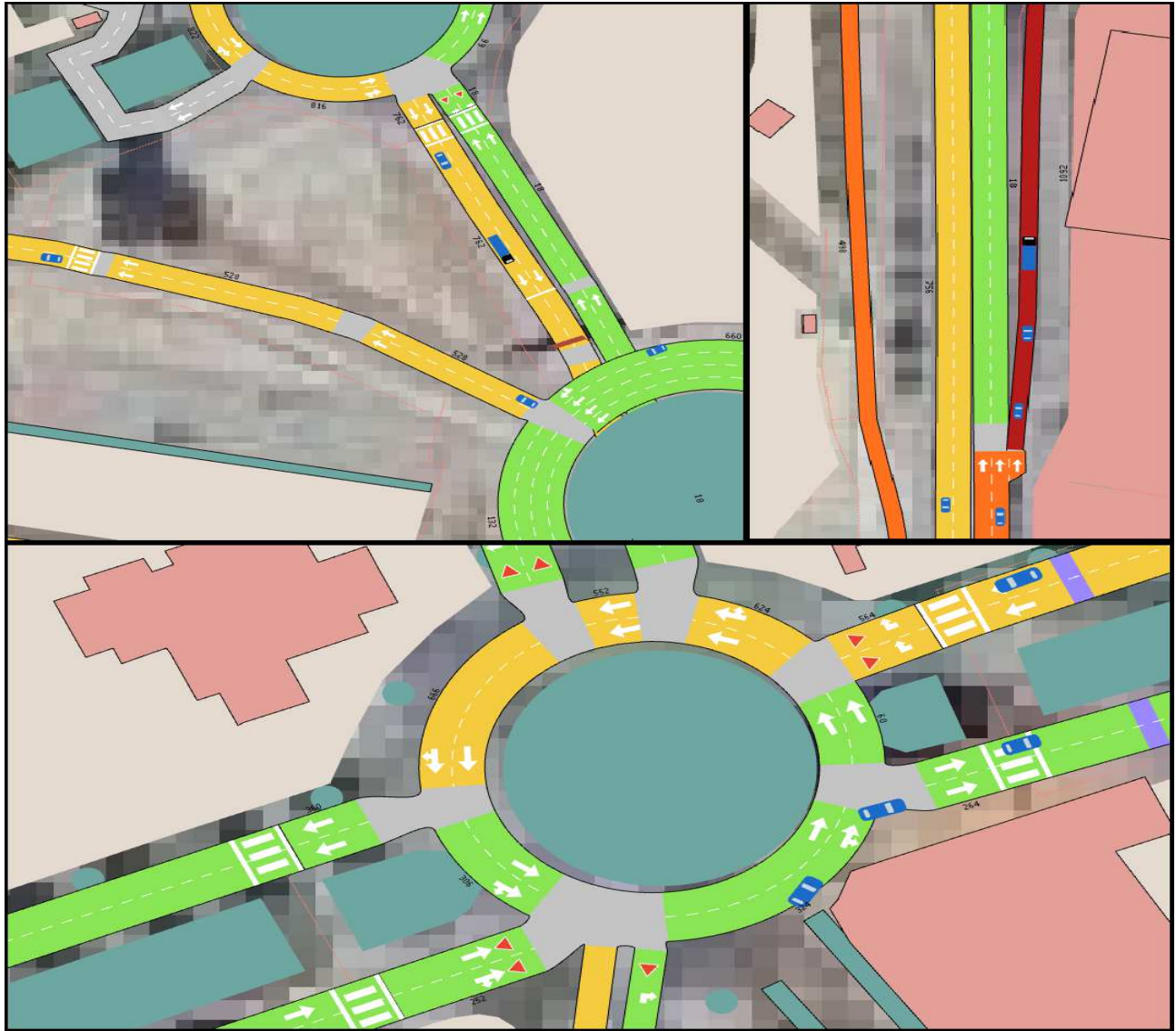


Figura 3.30. Detalle de flujo de vehículos en diferentes áreas.

En la parte derecha, dentro del recuadro inferior de figura 3.30, se observan dos franjas de color azul, dichas franjas son sensores repartidos por distintos puntos de la zona simulada, que ayudan a captar datos como velocidad, tiempo de paso entre un vehículo y el siguiente, distancia media recorrida y más datos menos relevantes. Gracias a estos sensores se puede realizar una serie temporal (figura 3.31) que analice la velocidad media de los vehículos. Con estos datos se podría calcular el EdC de los VE, con el algoritmo del apartado 3.1.2., introduciendo los datos de distancia recorrida y la velocidad media del VE.

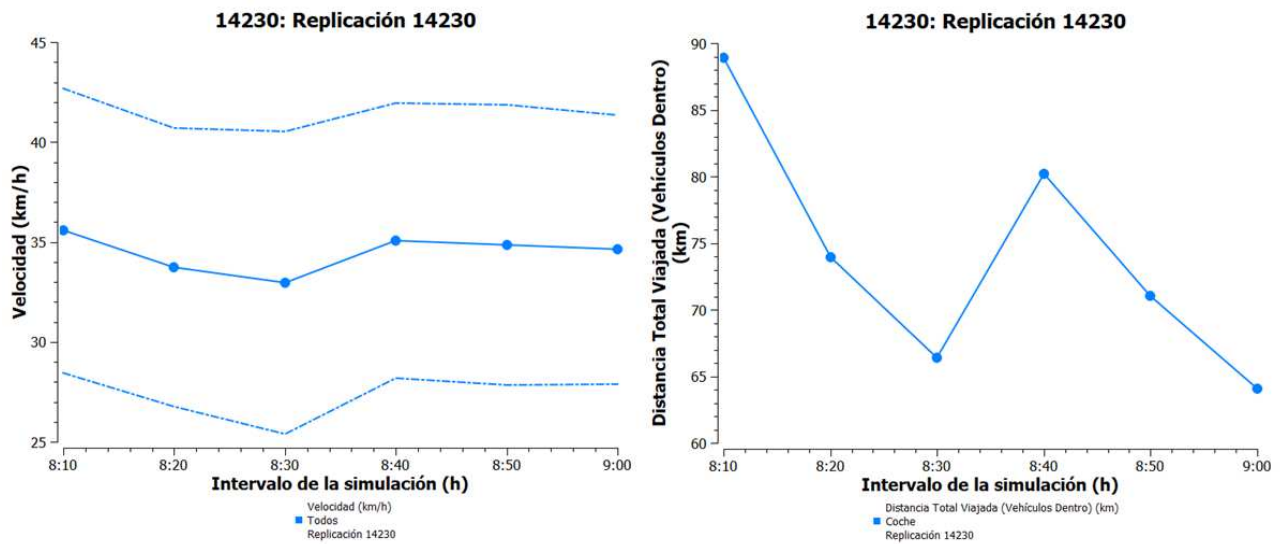


Figura 3.31. Velocidad y distancia recorrida por los vehículos.

Capítulo 4

Conclusiones, propuestas de mejora y aplicación práctica

Una vez completadas las simulaciones descritas en el capítulo 3, este capítulo expone una reflexión crítica sobre los resultados obtenidos y plantea una serie de soluciones y líneas de mejora para el despliegue efectivo de infraestructuras de carga de EVs en entornos urbanos. Además, se propone una aplicación práctica basada en los modelos desarrollados.

4.1. Conclusiones y propuestas de mejora

Como se explica anteriormente, la base del problema consta de dos puntos: la falta de EdC ya que las existentes son privadas, es decir, se ubican en aparcamientos subterráneos de pago, y la importancia que tiene el comportamiento de un conductor de VE. La aplicación de herramientas de simulación para la obtención de puntos clave (puntos donde la densidad de tráfico es mayor) donde posicionar nuevas EdC y poder abastecer la demanda de los VE, se ha tenido en cuenta como el núcleo de estudio en este Trabajo de Fin de Grado, obteniendo así, dos tipos de simulaciones muy distintas entre sí, utilizando dos herramientas diferentes, para el diseño de una serie de “soluciones”, lo más sólidas y realistas posibles, a dicho problema.

4.1.1. Influencia del (SoC) en el comportamiento del conductor

El análisis de las rutas simuladas con Python® demuestra que el comportamiento del conductor de EV se ve condicionado por el nivel de batería, especialmente en lo relativo a la planificación de paradas para recarga. A diferencia de los vehículos de combustión, los EVs requieren un seguimiento constante del SoC para garantizar la viabilidad del trayecto.

Propuesta: La integración de sistemas (como el propuesto en Python® en el capítulo 3) de navegación que informen en tiempo real sobre el SoC y propongan rutas optimizadas con paradas estratégicas en estaciones de carga (EdC) es clave para minimizar la “ansiedad de autonomía” y mejorar la experiencia de conducción de un VE.

4.1.2. Algoritmo de consumo personalizado por características del vehículo

Relacionado directamente con el punto anterior, a través del algoritmo implementado con Python®, se ha logrado estimar el consumo por kilómetro de cada EV en función de parámetros como el peso y la potencia motora. Aunque básico, el modelo resulta suficientemente robusto para establecer un primer análisis y clasificación de necesidades energéticas por tipo de vehículo.

Propuesta: Este algoritmo puede ampliarse incluyendo nuevas variables (velocidad media del trayecto, número de ocupantes, climatización, resistencia aerodinámica, tipo de vía, etc.), con el objetivo de personalizar aún más el cálculo de autonomía. Asimismo, puede integrarse como componente base en aplicaciones de gestión de rutas para VE. Datos como los de velocidad media y distancia de viaje, se pueden implementar directamente de la base de datos obtenida de la simulación con AimSun Next® (teniendo el alcance de una licencia sin restricciones, podría adaptarse a un sistema mucho más complejo).

4.1.3. Optimización de rutas con recarga intermedia

La simulación en dos tramos (inicio–EdC y EdC–destino) ha permitido evidenciar que la introducción de una parada intermedia no supone una penalización significativa en el recorrido si se realiza de forma óptima.

Propuesta: Integrar esta lógica en sistemas de navegación vehicular puede mejorar la eficiencia energética del trayecto, sobre todo si se considera la congestión del tráfico y los tiempos estimados de espera en cada estación. De hecho, complementando ambas simulaciones, se consiguen esos puntos de mayor congestión, donde podría existir una EdC y aprovechar esos momentos de “cola” para la recarga.

4.1.4. Localización estratégica de EdC según densidad y flujo de tráfico

Extendiendo el punto anterior, los resultados obtenidos con AimSun Next® muestran con claridad los puntos de mayor densidad y flujo de vehículos. Estas zonas coinciden con los momentos de mayor congestión, especialmente en el intervalo entre las 08:20 y las 08:40 horas, para esta simulación.

Propuesta: Ubicar las EdC en zonas de alta densidad vehicular maximiza su utilidad. Además, se propone aprovechar los momentos de congestión para recargar, ya que el tiempo perdido en espera puede destinarse a la recarga sin afectar significativamente al tiempo total de viaje, teniendo así una prevención para momentos futuros donde se necesite la recarga.

4.1.5. Limitación por falta de datos reales

Uno de los principales límites del estudio es la ausencia de datos reales sobre la movilidad de los EVs en Málaga. Esto impide una validación directa de los modelos y sus resultados. Aunque se haya diseñado lo más afín posible a la realidad.

Propuesta: Como futura mejora, se recomienda complementar este trabajo con datos reales obtenidos de operadores de estaciones de carga, registros municipales de tráfico o ampliar los datos obtenidos de plataformas como Electromaps [11]. Esto permitiría entrenar y ajustar los modelos de forma más precisa.

4.1.6. Combinación de herramientas de simulación como base metodológica

Finalmente, la combinación de Python® para la simulación de rutas individualizadas y AimSun Next® para la simulación de tráfico a gran escala ha permitido un análisis integral de la movilidad urbana de los VE. Ambas herramientas aportan perspectivas complementarias: detalle y personalización frente a visión agregada del comportamiento del tráfico.

Propuesta: Esta dualidad puede adoptarse como metodología estándar para estudios similares en otras ciudades, especialmente cuando se busca implementar políticas de movilidad sostenibles basadas en datos.

4.2. Aplicación práctica propuesta

A partir de los resultados de ambas simulaciones, se propone el diseño de una herramienta inteligente, como se prueba en [13], de apoyo a la conducción de EVs con las siguientes funcionalidades:

1. Cálculo predictivo del SoC en tiempo real durante la ruta, incorporando factores de tráfico y consumo personalizados.
2. Selección de EdC óptimas, considerando cercanía, ocupación prevista, tipo de cargador y tiempos estimados de recarga.
3. Optimización del momento de parada, priorizando los periodos de mayor congestión para reducir el impacto del tiempo de recarga en la duración total del trayecto.

Esta aplicación contribuiría a mejorar la eficiencia energética y la planificación del transporte eléctrico, facilitando la obtención de una curva de demanda energética dispuesta para nivelar los puntos extremos y conseguir una curva lo mas plana posible, además de la transición hacia una movilidad urbana más sostenible.

Referencias

- [1] Iglesias, E., “Petróleo y gas natural”. Economía actual, Ediciones Akal, 2003, <https://books.google.es/books?id=XHwqSWKseOoC>.
- [2] Lemburg, M.-A., “The python wiki,” 2025. <https://wiki.python.org/moin/> [Accedido: (14/05/2025)].
- [3] Developers, N., “Numpy, the fundamental package for scientific computing with python,” 2025. <https://numpy.org/> [Accedido: (14/05/2025)].
- [4] NumFOCUS, I., “Pandas,” 2025. <https://pandas.pydata.org/> [Accedido: (14/05/2025)].
- [5] Hunter, J., Dale, D., Firing, E., Droettboom, M., y development team, M., “Matplotlib,” 2025. <https://matplotlib.org/> [Accedido: (14/05/2025)].
- [6] Boeing, G., “Modeling and analyzing urban networks and amenities with osmnx,” Geographical Analysis, 2025, doi:10.1111/gean.70009. Publicado en línea antes de impresión.
- [7] Boeing, G., “Osmnx 2.0.3 documentation,” 2025. <https://osmnx.readthedocs.io/en/stable/> [Accedido: (15/05/2025)].
- [8] Hagberg, A. A., Schult, D. A., y Swart, P. J., “Exploring network structure, dynamics, and function using networkx,” en Proceedings of the 7th Python in Science Conference (Varoquaux, G., Vaught, T., y Millman, J., eds.), (Pasadena, CA USA), pp. 11–15, 2008.
- [9] Hagberg, A. A., Schult, D. A., y Swart, P. J., “Networkx, network analysis in python,” 2025. <https://networkx.org/documentation/stable/index.html> [Accedido: (15/05/2025)].
- [10] Foundation, O., “Openstreetmap,” 2025. <https://www.openstreetmap.org/#map=15/>

36.71434/-4.47624 [Accedido: (16/05/2025)].

- [11] Cañadell, X. y Rodríguez, J., “Electromaps,” 2025. <https://www.electromaps.com/es/puntos-carga/espana/malaga/malaga?page=3> [Accedido: (30/04/2025)].
- [12] Aimsun, Aimsun Next User Manual, s.f. Recuperado de <https://www.aimsun.com>.
- [13] Lopes, J. C., Sousa, T., y Melo, J. D., “Demand estimation for electric vehicle charging infrastructure: An extensive approach method,” *IEEE Access*, vol. 12, pp. 113215–113234, 2024, doi:10.1109/ACCESS.2024.3444735.

Anexo A

Resultados complementarios

En este apartado se muestran las tablas de resultados con AimSun Next®. En Tabla A.1 se observan cada uno de los resultados obtenidos con la simulación representada en el Capítulo 3.

Tabla A.1. Resumen completo de resultados de la simulación con AimSun

Serie Temporal	Valor	Desviación Estándar	Unidades
Cola Media - Todos	19.41	ND	veh
Cola Media - Coche	12.75	ND	veh
Cola Media - Camión	3.06	ND	veh
Cola Media - Motocicleta	3.6	ND	veh
Cola Virtual Máxima - Todos	41	ND	veh
Cola Virtual Máxima - Coche	29	ND	veh
Cola Virtual Máxima - Camión	8	ND	veh
Cola Virtual Máxima - Motocicleta	10	ND	veh

Continúa en la siguiente página

Serie Temporal	Valor	Desviación Estándar	Unidades
Cola Virtual Media - Todos	20.92	ND	veh
Cola Virtual Media - Coche	14.18	ND	veh
Cola Virtual Media - Camión	2.94	ND	veh
Cola Virtual Media - Motocicleta	3.79	ND	veh
Contaje de Entrada - Todos	4800	ND	veh
Contaje de Entrada - Coche	3308	ND	veh
Contaje de Entrada - Camión	579	ND	veh
Contaje de Entrada - Motocicleta	913	ND	veh
Densidad - Todos	6.82	ND	veh/km
Densidad - Coche	4.65	ND	veh/km
Densidad - Camión	0.95	ND	veh/km
Densidad - Motocicleta	1.23	ND	veh/km
Distancia Total de Viaje - Todos	5464.25	ND	km
Distancia Total de Viaje - Coche	3752.04	ND	km
Distancia Total de Viaje - Camión	678.54	ND	km
Distancia Total de Viaje - Motocicleta	1033.66	ND	km
Distancia Total Viajada (Vehículos Dentro) - Todos	92.39	ND	km

Continúa en la siguiente página

Serie Temporal	Valor	Desviación Estándar	Unidades
Distancia Total Viajada (Vehículos Dentro) - Coche	64.07	ND	km
Distancia Total Viajada (Vehículos Dentro) - Ca...	11.5	ND	km
Distancia Total Viajada (Vehículos Dentro) - Mo...	16.82	ND	km
Flujo - Todos	4668	ND	veh/h
Flujo - Coche	3216	ND	veh/h
Flujo - Camión	559	ND	veh/h
Flujo - Motocicleta	893	ND	veh/h
Flujo de Entrada - Todos	4800	ND	veh/h
Flujo de Entrada - Coche	3308	ND	veh/h
Flujo de Entrada - Camión	579	ND	veh/h
Flujo de Entrada - Motocicleta	913	ND	veh/h
Giros Perdidos - Todos	58	ND	
Giros Perdidos - Coche	11	ND	
Giros Perdidos - Camión	5	ND	
Giros Perdidos - Motocicleta	42	ND	
Número de Cambios de Carril - Todos	326.67	ND	#/km

Continúa en la siguiente página

Serie Temporal	Valor	Desviación Estándar	Unidades
Número de Cambios de Carril - Coche	193.28	ND	#/km
Número de Cambios de Carril - Camión	25.05	ND	#/km
Número de Cambios de Carril - Motocicleta	108.34	ND	#/km
Número de Paradas - Todos	0.06	ND	#/veh/km
Número de Paradas - Coche	0.06	ND	#/veh/km
Número de Paradas - Camión	0.06	ND	#/veh/km
Número de Paradas - Motocicleta	0.07	ND	#/veh/km
Número Total de Cambios de Carril - Todos	7758	ND	
Número Total de Cambios de Carril - Coche	4590	ND	
Número Total de Cambios de Carril - Camión	595	ND	
Número Total de Cambios de Carril - Motocicleta	2573	ND	
Número Total de Paradas - Todos	6805	ND	
Número Total de Paradas - Coche	4605	ND	
Número Total de Paradas - Camión	786	ND	

Continúa en la siguiente página

Serie Temporal	Valor	Desviación Estándar	Unidades
Número Total de Paradas - Motocicleta	1414	ND	
Tiempo de Demora - Todos	44.43	52.43	seg/km
Tiempo de Demora - Coche	41.81	38.57	seg/km
Tiempo de Demora - Camión	64.2	105.54	seg/km
Tiempo de Demora - Motocicleta	41.53	42.12	seg/km
Tiempo de Espera en Cola Virtual - Todos	15.57	56.49	seg
Tiempo de Espera en Cola Virtual - Coche	15.31	55.93	seg
Tiempo de Espera en Cola Virtual - Camión	18.2	61.66	seg
Tiempo de Espera en Cola Virtual - Motocicleta	14.84	55.12	seg
Tiempo de Espera Total (Esperando Fuera) - Todos	2.08	ND	h
Tiempo de Espera Total (Esperando Fuera) - Coche	1.55	ND	h
Tiempo de Espera Total (Esperando Fuera) - Camión	0.07	ND	h
Tiempo de Espera Total (Esperando Fuera) - Moto...	0.46	ND	h

Continúa en la siguiente página

Serie Temporal	Valor	Desviación Estándar	Unidades
Tiempo de Parada - Todos	19.38	49.54	seg/km
Tiempo de Parada - Coche	17.2	35.33	seg/km
Tiempo de Parada - Camión	32.75	102.74	seg/km
Tiempo de Parada - Motocicleta	18.87	40.12	seg/km
Tiempo de Viaje - Todos	112.41	53.16	seg/km
Tiempo de Viaje - Coche	110.06	39.09	seg/km
Tiempo de Viaje - Camión	134.99	106.27	seg/km
Tiempo de Viaje - Motocicleta	106.72	42.91	seg/km
Tiempo Total de Viaje - Todos	159.43	ND	h
Tiempo Total de Viaje - Coche	108.51	ND	h
Tiempo Total de Viaje - Camión	22.19	ND	h
Tiempo Total de Viaje - Motocicleta	28.73	ND	h
Tiempo Total de Viaje (Vehículos Dentro) - Todos	3.23	ND	h
Tiempo Total de Viaje (Vehículos Dentro) - Coche	2.2	ND	h
Tiempo Total de Viaje (Vehículos Dentro) - Camión	0.52	ND	h
Tiempo Total de Viaje (Vehículos Dentro) - Moto...	0.52	ND	h

Continúa en la siguiente página

Serie Temporal	Valor	Desviación Estándar	Unidades
Vehículos Dentro - Todos	132	ND	veh
Vehículos Dentro - Coche	92	ND	veh
Vehículos Dentro - Camión	20	ND	veh
Vehículos Dentro - Motocicleta	20	ND	veh
Vehículos Esperando para Entrar - Todos	34	ND	veh
Vehículos Esperando para Entrar - Coche	25	ND	veh
Vehículos Esperando para Entrar - Camión	2	ND	veh
Vehículos Esperando para Entrar - Motocicleta	7	ND	veh
Vehículos Fuera - Todos	4668	ND	veh
Vehículos Fuera - Coche	3216	ND	veh
Vehículos Fuera - Camión	559	ND	veh
Vehículos Fuera - Motocicleta	893	ND	veh
Vehículos Perdidos Dentro - Todos	0	ND	veh
Vehículos Perdidos Dentro - Coche	0	ND	veh
Vehículos Perdidos Dentro - Camión	0	ND	veh

Continúa en la siguiente página

Serie Temporal	Valor	Desviación Estándar	Unidades
Vehículos Perdidos Dentro - Motocicleta	0	ND	veh
Vehículos Perdidos Fuera - Todos	0	ND	veh
Vehículos Perdidos Fuera - Coche	0	ND	veh
Vehículos Perdidos Fuera - Camión	0	ND	veh
Vehículos Perdidos Fuera - Motocicleta	0	ND	veh
Velocidad - Todos	34.41	7.11	km/h
Velocidad - Coche	34.64	6.77	km/h
Velocidad - Camión	30.3	7.03	km/h
Velocidad - Motocicleta	36.15	7.38	km/h
Velocidad Harmónica - Todos	32.03	8.74	km/h
Velocidad Harmónica - Coche	32.71	7.95	km/h
Velocidad Harmónica - Camión	26.67	9.84	km/h
Velocidad Harmónica - Motocicleta	33.73	9.03	km/h