

|  |
| --- |
|  |

|  |
| --- |
| **Trabajo Fin de Máster** |
| **Electric Route** |
| **Blanca Alonso González**  **Marta García Palomo**  **Irene Robles Cabezas**  **Lucía Tomaino De la Cruz** |

Contenido

[1. Introducción y Objetivos 2](#_Toc62920999)

[1.1. Estructura del trabjo 2](#_Toc62921000)

[1.2. Herramientas Software 2](#_Toc62921001)

[2. Comprensión de los datos 4](#_Toc62921002)

[2.1. Coches eléctricos 4](#_Toc62921003)

[2.2. Ciudades 4](#_Toc62921004)

[2.3. Puntos de recarga 4](#_Toc62921005)

[2.4. Gasolineras 5](#_Toc62921006)

[3. Modelo: definición y restricciones 6](#_Toc62921007)

[3.1. Cálculo de la matriz de distancias 6](#_Toc62921008)

[3.2. Cálculo de ruta óptimo entre dos puntos 6](#_Toc62921009)

[3.3. Definición de restricciones 6](#_Toc62921010)

[3.4. Modelo completo 6](#_Toc62921011)

[4. Herramienta web: Flask 7](#_Toc62921012)

[5. Docker 8](#_Toc62921013)

[6. Discusión y conclusiones 9](#_Toc62921014)

[7. Próximos pasos 9](#_Toc62921015)

[8. Referencias 9](#_Toc62921016)

[9. Anexo. 9](#_Toc62921017)

[9.1. Función 9](#_Toc62921018)

# Introducción y Objetivos

Durante los últimos años se ha producido un incremento en las ventas de coches eléctricos respecto a datos anteriores. En Noviembre 2020, Sara Aagesen, secretaria de Estado de Energía, informaba que en una comparecencia en la Comisión de Transición Ecológica y Reto Demográfico del Congreso de los Diputados, que el Gobierno ha incluido dentro del plan de recuperación que ha remitido a la Unión Europea una partida de 1.100 millones destinados a la movilidad eléctrica y que podría activarse ya en 2021.

Debido al aumento de las ventas y el impulso a la movilidad con este tipo de vehículos, surge la motivación de realizar este Trabajo de Fin de Máster dónde el principal objetivo consiste en desarrollar un recomendador de rutas para coches eléctricos según su autonomía. De forma que, dados dos puntos de la Península Ibérica, origen y destino, y, seleccionado un tipo de coche eléctrico, se recomendará una ruta óptima en distancia y tiempo para ese recorrido. A la hora de determinar la ruta se tendrá en cuenta la autonomía del coche, de manera que si el viaje está limitado por la batería del coche se le recomendará al usuario un punto de recarga en medio del recorrido.

Además, es posible que durante la ruta no exista un punto de recarga conveniente para el coche eléctrico en cuestión, ya que los puntos de recarga pueden tener un tipo de conector no compatible. En este caso, se recomendará un punto de recarga ficticio, situado en alguna gasolinera. Con esta recomendación se puede ver dónde sería interesante instalar nuevos puntos de recarga.

Para el desarrollo de todo el trabajo se utiliza un repositorio para control de versiones dónde se puede consultar todo el código del trabajo [[github](https://github.com/luciatomainodelacr/TFM)].

## **Estructura del trabajo**

Este trabajo se estructura de la siguiente forma. En el primer apartado [[1](#_Introducción_y_Objetivos)], se hace una introducción al trabajo planteado y la motivación del mismo, Además, se incluye una breve descripción de todas las herramientas de software utilizadas a lo largo del trabajo. En el segundo apartado [[2](#_Comprensión_de_los_1)] se describen los distintos conjuntos de datos con los que se va a trabajar, el origen de los mismos, las variables que los componen y las modificaciones o tratamiento que se ha realizado sobre ellos.

\*\*\*\*\* rellenar al completar la memoria

## **Herramientas Software**

Para el desarrollo de trabajo se utilizan diferentes softwares que se explican brevemente a continuación:

* **Github**: herramienta web para la gestión de repositorios basada en Git. Permite almacenar y gestionar código, además de llevar un control de versiones durante el desarrollo del trabajo. Se utiliza como repositorio del proyecto completo, incluyendo los conjuntos de datos, los script utilizados para la obtención y depuración de los datos, el desarrollo del modelo, el despliegue de la web y la creación de contenedores de aplicaciones.
* **MySQL**: es un sistema de gestión de bases de datos relacional. Se utiliza para gestionar las diferentes bases de datos con las que se trabaja.
* **Python**: lenguaje de programación elegido.
* **Flask**: es un microframework paraPython que permite crear aplicaciones web. Para el desarrollo de la aplicación web se hace uso de Flask, incluyendo otros lenguajes de programación necesarios como html, javascript o css.
* **Grafana**: es un software libre basado en licencia de Apache, que permite la visualización y el formato de datos métricos. Se utiliza para \*\*\*\*\*.
* **Docker**: \*\*\*\*\*
* **Google Cloud Platform**: es una plataforma de computación en la nube que ofrece diversos servicios. Se utiliza para \*\*\*\*\*.

Además de estos softwares se hace uso de algunas APIs de Google o de Tomtom para obtener información. Estas aplicaciones son las siguientes:

* **Geocoding Service:** Se utiliza para \*\*\*\*\*.
* **Distance Matrix:** Se utiliza para \*\*\*\*\*.
* **Tomtom – EV Charging Stations Availability:** Se utiliza para \*\*\*\*\*.

# Comprensión de los datos

El conjunto de datos con el que se trabaja consta de varias bases de datos, por un lado se tiene una base de datos para los distintos tipos de coches eléctricos. Por otro lado, una base de datos que recoge los distintos puntos de recarga existentes en la Península Ibérica. Finalmente, como se ha introducido en los objetivos, se obtiene una base de datos referente a las gasolineras

## **Coches eléctricos**

Origen del datset. Limpieza del código, eda, etc. Variables que lo componen.

El listado de coches eléctricos que se va a utilizar está publicada en página de Kaggle, [[EVs – One Electric Vehicle Dataset – Smaller](https://www.kaggle.com/geoffnel/evs-one-electric-vehicle-dataset)] y, está compuesta por dos ficheros: *ElectricCarData\_Clean.csv* y *ElectricCarData\_Norm.csv*. La única diferencia entre ambos es que en el fichero *\_Norm* las variables numéricas contienen las unidades para cada observación, y en el fichero *\_Clean* se han eliminado y renombrado las variables. Se utilizará por tanto el fichero: *ElectricCarData\_Clean.csv.*

Limpieza del código, eda, etc

El conjunto de datos definitivo tiene un total de 104 observaciones y 8 variables que se describen a continuación:

* Brand: marca del coche. Tipo de dato: string.
* Model: modelo del coche. Tipo de dato: string.
* Range\_km: autonomía del coche en km. Tipo de dato:
* Efficiency\_whkm: consumo del coche por cada 100 km. Tipo de dato:
* Fastcharge\_kmh: velocidad de carga rápida. Tipo de dato:
* RapidCharge: flag que toma el valor *Yes* si el coche es de carga rápida, *No* en caso contrario.
* Plugtype: tipo de conector compatible con el coche. Tipo de dato:
* Battery\_capacity:

## **Ciudades**

API google obtener listado puntos de recarga.

De cara a definir los puntos de origen y destino para las distintas rutas que pueda realizar el usuario, se decide construir una base de datos de las principales ciudades de España tomando como puntos de referencia las estaciones de tren y autobús. Una vez elabadora la lista, mediante la Geocoding API de Google, se geolocalizan las coordenadas geográficas de cada punto. Este conjunto de datos tiene un total de 93 observaciones y 6 variables que se describen a continuación:

* id: código único asociado a cada punto. Está formado por la ciudad y el tipo de estación que sea (tren o bus). Tipo de dato: string.
* Provincia: ciudad a la que corresponde el punto. Tipo de dato: string.
* Direccion: indica si la estación es de autobuses o de tren. Tipo de dato: string.
* Latitud: latitud del punto. Tipo de dato: float64.
* Longitud: longitud del punto. Tipo de dato: float64.
* Coordenadas: tupla que contiene los valores de la latitud y la longitud. Tipo de dato: object.

## **Puntos de recarga**

API google obtener listado puntos de recarga. Reducir puntos de recarga. Obtener información desde la API de Tomtom. Variables que lo componen.

El conjunto de datos definitivo tiene un total de 353 observaciones y 11 variables que se describen a continuación:

* id: código único asociado a cada punto de recarga. Tipo de dato: string.
* Latitude:
* Longitude:
* Name:
* StreetName:
* Provincia:
* CCAA:
* PostalCode:
* ConnectorType:
* RatedPowerKW:
* Num\_connectors:

## **Gasolineras**

Obtener dataset. Reducir puntos de recarga. Variables que lo componen.

Este conjunto de datos tiene un total de 332 observaciones y 6 variables que se describen a continuación:

* id: código único asociado a cada gasolinera. Tipo de dato: string.
* Provincia: provincia a la que pertenece la gasolinera. Tipo de dato: string.
* Municipio: municipio a la que pertenece. Tipo de dato: string.
* Localidad: localidad a la que pertenece. Tipo de dato: string.
* Codigo\_postal: código postal de la gasolinera. Tipo de dato: float64.
* Dirección: dicción de la gasolinera. Tipo de dato: string.

# Modelo: definición y restricciones

El

## **Cálculo de la matriz de distancias**

Pruebas API Google Distance Matrix. Pricing. Función Haversine. Ejemplo

## **Cálculo de ruta óptimo entre dos puntos**

Librería networkx. Algoritmo de Dijkstra. Camino más corto según KPI. Ejemplo

## **Definición de restricciones**

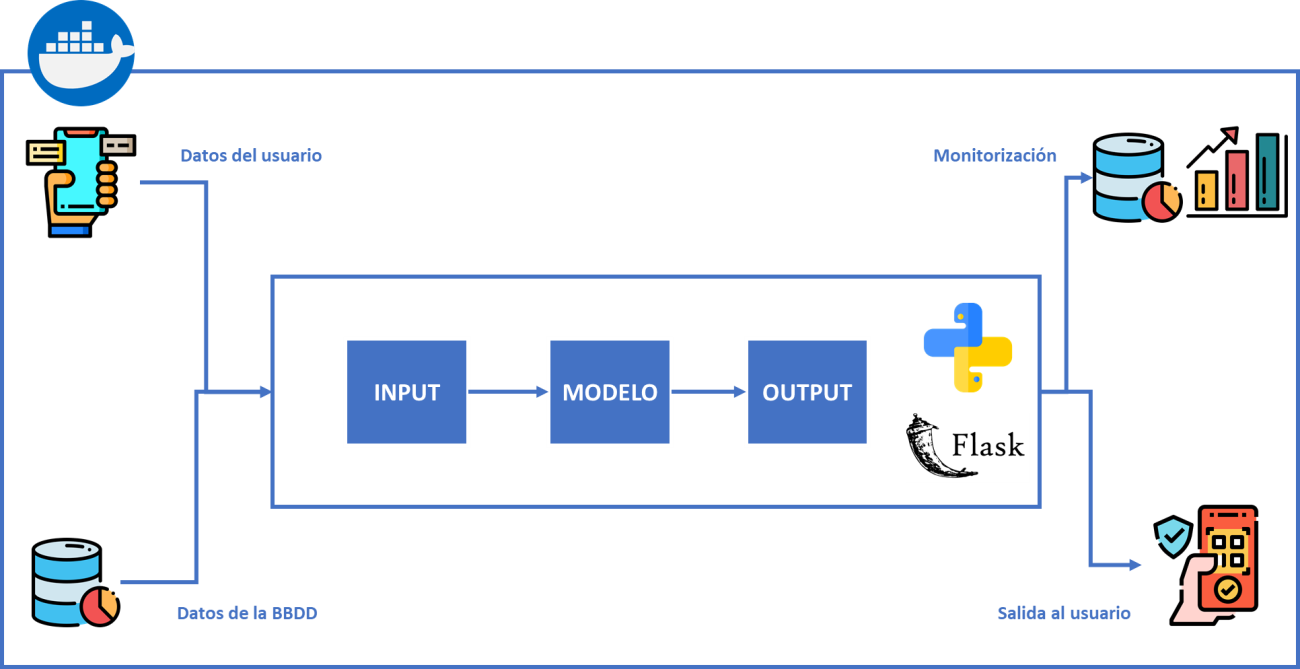
Explicación diferentes restricciones. de filtrado. Lógica que se construye

## **Modelo completo**

Explicación y lógica con la que se construye.

# Herramienta web: Flask

El diseño funcional de la aplicación es:



# Productivicación

El

## **Docker**

## **Google Cloud Platform**

# Discusión y conclusiones

El

# Próximos pasos

El

# Referencias

El

# Anexo.

En el siguiente anexo se incluye algunas funciones propias o facilitadas por otros profesores que se han utilizado durante el desarrollo del trabajo.

## Función

Esta función se utiliza en el apartado [\*\*] para