



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



DEPARTAMENTO DE SISTEMAS  
INFORMÁTICOS Y COMPUTACIÓN

Departamento de Sistemas Informáticos y Computación  
Universitat Politècnica de València

# **Optimización de mapas de consumo de combustible en vehículos eléctricos.**

**TRABAJO FIN DE MÁSTER**

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

*Autor:* Josep Vicent Dols Dart

*Tutor:* Antonio Garrido Tejero

Curso 2019-2020



# Resumen

**Palabras clave:** coche, eléctrico, batería, estimación, ruta

---

# Abstract

**Key words:** car, electrical, battery, estimate, route

---



# Índice general

---

<b>Índice general</b>	<b>V</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>Índice de tablas</b>	<b>VII</b>

---

<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	1
1.2 Objetivos del proyecto . . . . .	1
1.3 Estructura . . . . .	2
<b>2 Estado del arte</b>	<b>3</b>
2.1 Crítica . . . . .	3
2.2 Propuesta . . . . .	4
<b>3 Análisis del problema</b>	<b>5</b>
3.1 Identificación de posibles soluciones . . . . .	6
3.2 Solución propuesta . . . . .	6
<b>4 Diseño</b>	<b>7</b>
4.1 Arquitectura del sistema . . . . .	7
4.2 Diseño detallado . . . . .	7
4.3 Tecnología utilizada . . . . .	8
<b>5 Desarrollo</b>	<b>9</b>
5.1 Obtención de los datos . . . . .	9
5.2 Modelo de consumo del vehículo . . . . .	11
<b>6 Implantación</b>	<b>13</b>
<b>7 Pruebas</b>	<b>15</b>
<b>8 Conclusión</b>	<b>17</b>
8.1 Opciones futuras de líneas de desarrollo . . . . .	17
<b>Bibliografía</b>	<b>17</b>

---

Apéndice	
<b>A Glosario</b>	<b>21</b>



## Índice de figuras

---

## Índice de tablas

---





---

---

# CAPÍTULO 1

## Introducción

---

Gracias a los avances tecnológicos que se están produciendo en la actualidad en diferentes ámbitos de investigación, como pudieran ser el diseño de baterías eléctricas con mayor capacidad o las mejoras en el campo de la automoción, introduciendo elementos como dichas baterías o una gran cantidad de elementos relacionados con el confort y seguridad de los pasajeros de los vehículos, este sector está evolucionando hacia una nueva forma de movilidad, actualmente observable en la comercialización de los vehículos híbridos, y tal y como indican todas las previsiones, y con el interés del ámbito político, en un futuro cercano, en vehículos eléctricos.

Uno de los principales problemas que existen actualmente dentro del ámbito de los vehículos eléctricos es la capacidad de carga de las baterías que se usan en los mismos, las cuales están aún en proceso de evolución tecnológica; esto se traduce en la realidad, en que la autonomía de estos vehículos no es aún equiparable a la de los vehículos de combustión tradicionales o a los más recientes vehículos híbridos.

### 1.1 Motivación

---

Teniendo en cuenta las limitaciones y el estado actual de la tecnología implementada en estos vehículos eléctricos, uno de los puntos más importantes a considerar es la optimización del estilo de conducción durante el transcurso de una ruta, ayudando a la reducción del consumo eléctrico de la batería durante la misma. Técnicamente, este tipo de cálculos de optimización conlleva una gran cantidad de recursos computacionales asociados, especialmente si lo que se busca es conseguir es la ruta óptima.

Para el desarrollo que se propone, lo que se busca conseguir es el cálculo de una ruta, que sea la mejor posible, pero sin la necesidad de que esta sea la óptima. Para conseguir esto, se busca utilizar los llamados algoritmos genéticos, ya que estos proporcionan un gran rango de posibilidades y mucha flexibilidad para un problema con muchos factores a considerar a la hora de realizar los cálculos necesarios para poder alcanzar unos resultados válidos. A su vez, esta aproximación permite con relativa facilidad adaptarse a nuevos requisitos o condiciones para calcular nuevas rutas en diferentes situaciones.

### 1.2 Objetivos del proyecto

---

El objetivo principal de este proyecto es el desarrollo de una aplicación de escritorio, capaz de analizar una ruta de carretera dada por un punto de origen y un punto de destino introducidos por el usuario y dar como resultado de salida a dicho usuario una

serie de puntos indicando en cada uno de ellos la aceleración o frenada que tiene que aplicar un vehículo eléctrico para seguir la mejor ruta posible a realizar en términos de eficiencia energética.

Dentro de ese objetivo global, hay una serie de objetivos secundarios o internos a cumplir durante el desarrollo para garantizar una funcionalidad básica dentro de la propia aplicación:

- Crear diferentes perfiles de optimización; principalmente, uno basado en la optimización temporal de la ruta, donde se prime ante todo tardar lo menos posible en recorrerla, y otro basado en la optimización de los recursos (capacidad de carga en este caso), donde se busca intentar consumir la menor carga de batería posible.
- Permitir aplicar todo el proceso de optimización con diferentes vehículos eléctricos que ya se encuentren en el mercado, y en casos más específicos, con diferentes cargas dentro de un mismo vehículo.
- Permitir también realizar el proceso de simulación con diferentes condiciones climáticas, con el efecto que se le puede atribuir a cada una de ellas.

### 1.3 Estructura

---

En cuanto a la estructura y distribución del proyecto, dada la naturaleza del mismo y el uso de algoritmos genéticos que se hace para la resolución del problema, este se divide en dos bloques principales.

El primer bloque consiste en la definición y explicación de la aplicación que se va a crear, a modo de interfaz con el usuario, del proceso seguido y de las validaciones realizadas para comprobar que los datos que se introducen en la misma y que se van a utilizar en el proceso de simulación son válidos y correctos.

En cuanto al segundo bloque, este se basa en las pruebas de ajuste, o *tuning*, a realizar sobre el modelo del algoritmo genético y las diferentes funciones de coste utilizadas para obtener los resultados esperados. Al final de este bloque, se encontrarán todos los resultados obtenidos durante el proceso, junto a una comparación de los mismos.

---

## CAPÍTULO 2

# Estado del arte

---

En cuanto al estado del arte relacionado con este proyecto, existen diferentes aproximaciones dependiendo del aspecto a valorar. Por un lado se encuentra el campo relacionado con la automoción, y de forma más específica con los vehículos eléctricos. De otro lado, también se encuentra el campo relacionado con las técnicas utilizadas para resolver el problema, que en este caso se trata de un algoritmo genético.

Sin embargo, a la hora de buscar aproximaciones a la unión de ambos campos no se pueden encontrar una gran cantidad de publicaciones al respecto. Así, la gran mayoría de estas se centran en la resolución de un problema bastante similar, en cuanto al cálculo de rutas o planificación temporal, pero centrada en vehículos de tecnología híbrida. Esto se puede apreciar en publicaciones como [1], [5] o [6]. En las dos primeras publicaciones se aprecia como se intenta resolver el problema de consumo de energía (o búsqueda de una estrategia para mejorarlo) aplicando principalmente técnicas basadas en la programación dinámica.

Pese a centrarse en vehículos híbridos, es muy conveniente observar como los modelos utilizados para simular el comportamiento de un vehículo son prácticamente idénticos entre las diferentes publicaciones, coincidiendo también con el modelo propuesto en [4], siendo este un modelo para un vehículo puramente eléctrico.

En cuanto a la herramienta de simulación, en este caso basada en algoritmos genéticos, se trata de un campo con una gran variabilidad de propuestas y de gran versatilidad, donde centenares de modelos diferentes se han creado para resolver problemas totalmente distintos, por lo que no todas las técnicas se pueden aplicar favorablemente ante un problema como el que se busca resolver aquí. Para este problema en específico, cabe destacar la aproximación propuesta en [3], donde se potencia mucho más la parte mutacional del algoritmo que los cruces iniciales.

### 2.1 Crítica

---

Como se ha dejado entrever en el análisis realizado sobre la situación actual en la que se encuentran los diferentes campos de investigación y desarrollo relacionados o con aplicación directa a este proyecto, aún no se ha profundizado en la planificación de rutas para vehículos puramente eléctricos.

Pese a que es un campo naciente, y donde es difícil encontrar estándares que permitan comparar las diferentes aproximaciones que existen en cuanto a términos de modelos de simulación; esto se debe en gran parte al factor de la industria automovilística, donde cada fabricante dispone de sus propios modelos de simulación, pero sin publicarlos abiertamente. Es por ello, que los modelos existentes (y a partir de los que se trabaja) son

aproximaciones realizadas (y en muchos casos muy simplificadas) del comportamiento de un motor eléctrico de verdad.

## 2.2 Propuesta

---

En este proyecto se busca conseguir un modelo de simulación o predicción de rutas que combine lo mejor posible las diferentes técnicas existente actualmente de dos campos diferentes como son la automoción y la inteligencia artificial. Se trata en este caso de combinar los modelos que existan de simulación de motores eléctricos junto con diferentes técnicas de algoritmos genéticos, adaptando ambos recursos para obtener una predicción lo más acertada posible.

---

## CAPÍTULO 3

# Análisis del problema

---

Como ya se ha indicado anteriormente, el propósito de este proyecto se basa en la creación de una aplicación destinada a la planificación de una ruta a ser realizada por un vehículo eléctrico. Como se detallará más adelante, para ello se va a crear dicha aplicación desde el inicio, sin reutilizar componentes ya existentes.

Dado que esta aplicación se trata de un ejecutable disponible para diferentes plataformas, se plantean una serie de requisitos o elementos mínimos a definir y desarrollar para garantizar una usabilidad básica que acompañe a los resultados que se puedan obtener. Entre estos requisitos, destacan:

- Permitir la elección de diferentes vehículos con los que realizar la predicción. Estos vehículos deberían de encontrarse entre un rango el cual alcance el mayor segmento de opciones dentro de los vehículos de modo eléctrico.
- Permitir también que el usuario sea capaz de modificar las condiciones climáticas de la ruta, viéndose afectada la predicción por estos cambios.
- El usuario introducirá la posición tanto de origen como de destino a partir de sus coordenadas geográficas.

Además de estos requisitos relacionados principalmente con la interfaz de la aplicación, existe un segundo conjunto relacionado con la funcionalidad de la misma, es decir, con el algoritmo de predicción a utilizar.

- En todo momento se deben de respetar los límites de velocidad de las vías a través de las cuales se establece que debería de circular el vehículo.
- Además del límite de las vías, otros factores como la pendiente o incluso la superficie puede ser tomados en cuenta para realizar la predicción.

Finalmente, un último requisito que combina a los dos bloques anteriores se trata de lo siguiente:

- Debe de ser posible seleccionar el modo de priorización de la ruta, es decir, elegir que indicador(energía consumida, tiempo empleado, etc.) se prefiere mejorar respecto a otros o intentar obtener una predicción que combine varios.

### 3.1 Identificación de posibles soluciones

---

Ante los diferentes retos que se aprecian para tratar de cumplir con los objetivos mencionados anteriormente, aparece un abanico de posibilidades relacionadas con la mejor forma de abordar el problema a tratar. Todo este conjunto de posibilidades se pueden englobar dentro del campo de los métodos metaheurísticos.

Dentro de este campo de técnicas metaheurísticas, existen varias de ellas a tener en cuenta a la hora de intentar proponer una solución al problema planteado previamente. Algunas de estas técnicas que se han tenido en cuenta son: PSO (*Particle Swarm Optimization* u optimización por enjambre de partículas), *hill climbing* (Algoritmo de escalada simple), SA (*Simulated Annealing* o enfriamiento simulado) y GA (*Genetic Algorithms* o algoritmos genéticos).

Todas estas técnicas comparten el factor común de que las soluciones que ofrecen son los llamados máximos locales, es decir, que dicha solución puede o puede no ser la mejor posible dado unos datos de entrada. Otro factor importante que se ha tenido en cuenta es la facilidad que aportan las diferentes técnicas para adaptarse al problema y minimizar lo máximo posible la complejidad de la aplicación a desarrollar.

### 3.2 Solución propuesta

---

Teniendo en cuenta las diferentes técnicas mencionadas, se ha optado por elegir GA (algoritmos genéticos) para intentar resolver el problema actual. Adicionalmente a los diferentes factores ya mencionados, cabe destacar que esta aproximación aporta una gran flexibilidad a la hora de escoger tanto la plataforma como las tecnologías (lenguajes y entornos de programación) a utilizar, ya que su implementación no viene forzada por ningún patrón marcado o estandarizado y a su vez, permite adaptar las diferentes características de dicho método a las exigencias específicas del problema.

A la hora de aplicar estas técnicas, se va a contar en todo momento con una serie de restricciones, basadas en el entorno del mundo real, ya que las soluciones a encontrar tienen como objetivo su aplicación en dichos entornos reales. Dentro de estas restricciones, se encuentran elementos como los límites de velocidad de las calzadas, dependiendo de los tramos o carreteras, los límites en cuanto a prestaciones de un vehículo eléctrico, los límites aplicados por el medio (estado meteorológico, o pendiente y peralte de un tramo de carretera) y finalmente, los propios límites impuestos por el usuario en cuanto a preferencias personales.

Para poder tener en cuenta estas restricciones, junto al desarrollo de la aplicación y la creación del algoritmo personalizado, se van a realizar una serie de pruebas adicionales para encontrar los puntos o regiones límites de la simulación y como se debería de comportar en dichos puntos o que máximos o mínimos de ciertos aspectos se debería de requerir a las diferentes partes de las soluciones propuestas

---

## CAPÍTULO 4

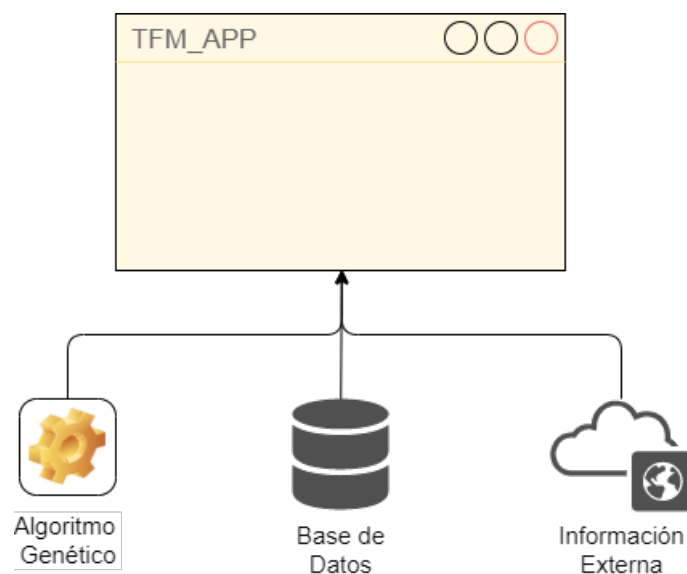
# Diseño

---

### 4.1 Arquitectura del sistema

---

El sistema a desarrollar se va a centrar principalmente en una única aplicación de escritorio, que será la encargada de lanzar las diferentes ejecuciones del algoritmo genético y de mostrar visualmente la mejor solución obtenida. Dentro de esta aplicación, hay tres elementos diferentes a mencionar y que conforman los diferentes bloques de la misma. Dichos bloques se tratan de la propia lógica del algoritmo, la gestión de la información de los vehículos y la obtención de la información de la ruta a realizar. En este caso, la información de los vehículos se introduce de forma estática en la propia aplicación, sin necesidad de agentes externos. Un posible diagrama de estas relaciones sería:

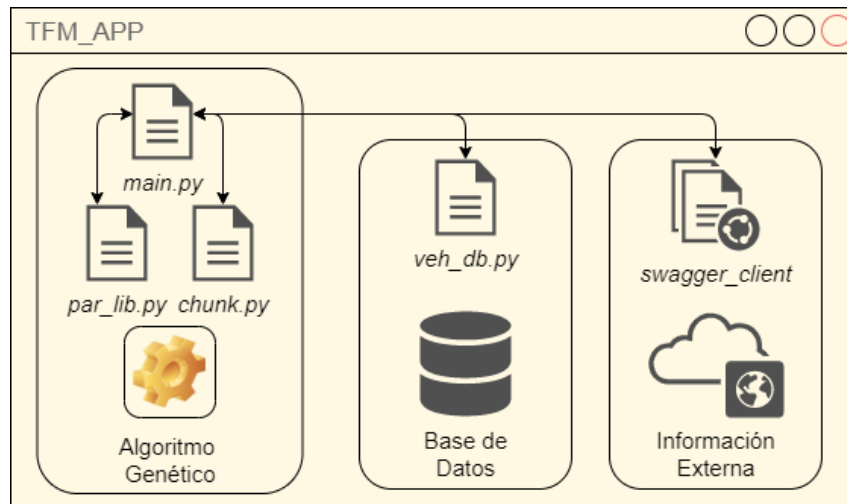


### 4.2 Diseño detallado

---

Entrando en un poco más de detalle sobre la arquitectura definida en el punto anterior, nos vamos a encontrar con una aplicación formada por una clase principal (*main.py*) donde se encuentra la mayoría de la lógica a utilizar durante la ejecución del algoritmo genético. Alrededor de esta clase principal, se han definido una serie de elementos auxiliares, a modo de objetos, que ayudan a la implementación de la funcionalidad del programa y a su vez incluyen algunas funciones a ser utilizadas sobre dichos objetos, estos serían *par\_lib.py* y *chunk.py*.

Para la definición de la base de datos de los vehículos, se ha utilizado también un fichero suelto (*veh\_db.py*) en el cual se incluye toda la información a ser utilizada en el resto de módulos de la aplicación. Finalmente, la obtención de la información queda dividida entre una parte del programa principal y un último fichero externo que hace las funciones de interfaz hacia el exterior. Siguiendo el esquema del punto anterior, un nuevo enfoque al mismo quedaría así:



### 4.3 Tecnología utilizada

Para llevar a cabo todo este diseño mencionado anteriormente, se va a utilizar como lenguaje de programación Python 3, específicamente la versión 3.6.8. Esta elección se justifica por diversos motivos. El primero, dado a la gran cantidad de herramientas y soporte que existe para este lenguaje y la facilidad que aporta el mismo para poder realizar diferentes implementaciones y que estas no se vean limitadas por el factor tecnológico.

Un segundo punto a tener en cuenta se trata del factor multiplataforma, ya que esta elección permite que el desarrollo se haga de forma completamente idéntica desde plataformas Windows, Linux o MacOS. El tercer elemento a considerar se trata del hecho de que este lenguaje ya incorpora una librería, Tk, para realizar la gestión de las ventanas de la aplicación, junto a *matplotlib*, la librería encargada de mostrar las diferentes gráficas del algoritmo a través de dichas ventanas.

Para la obtención de la información relacionada con las rutas sobre las que aplicar el algoritmo, se ha utilizado la web Graphhopper, la cual dispone de una API pública, a la que se puede acceder a través de una librería de Python (entre otros lenguajes) disponible en la web oficial y sobre la que se realizan todas las peticiones.

Finalmente, destacar que para realizar la codificación de dicha aplicación se ha trabajado a través del editor Visual Studio Code de Microsoft, aprovechando sus herramientas disponibles para Python de auto-completado y de depuración.



---

## CAPÍTULO 5

# Desarrollo

---

### 5.1 Obtención de los datos

---

Como ya se ha mencionado anteriormente, la herramienta a partir de la cual se han ido obteniendo los datos para realizar las estimaciones posteriores se trata de Graphhopper. Esta página web ofrece un servicio principal similar al de otras páginas como Google Maps o Bing Maps; sin embargo, la utilidad principal que se va a aprovechar de esta se basa en el uso de la API (*Application Programming Interface*) que dispone, en especial de la denominada como *Routing API*.

A partir de una llamada a dicha API, que se detalla a continuación, es posible obtener la ruta que debería de seguir un vehículo para llegar desde un punto A, marcado por sus coordenadas geográficas, hasta un punto B, también marcado por sus coordenadas. Esta ruta viene enmarcada dentro de un objeto JSON, donde, entre otros elementos, se encuentra la información relativa a la distancia total de la ruta, una estimación de tiempo y algunos otros elementos de utilidad para la posible representación gráfica de la ruta.

En cuanto a la propia ruta en si, esta viene marcada por una lista de elementos, (points), donde cada uno de estos elementos se corresponden a una coordenada geográfica (o *punto*, por tal y como se referencia en la documentación de la API), incluyendo esta su latitud, longitud y altitud (en metros) sobre el nivel del mar. Es a partir de este tercer campo desde donde se obtiene el perfil de la ruta a procesar.

Adicionalmente, se obtienen las listas *max\_speed* y *road\_class*, donde se indica la velocidad máxima a aplicar en los segmentos que hay entre los dos puntos que se indican de la lista de puntos, de la misma forma que se indica el tipo de carretera. Así, es posible estimar para todos los segmentos el límite a seguir, bien establecido directamente o bien inferido del tipo de carretera a utilizar.

A continuación se muestra la petición HTTP que se envía a la API para realizar la consulta de la ruta entre dos puntos, A (39.462160,-0.324177) Y B (39.441699,-0.595555), e indicando que esta se va a realizar en un coche (o vehículo de peso menor a 3500 Kg). Además de estos parámetros básicos referentes a la ruta en si, también se añaden diferentes opciones para obtener la información relativa a la altitud de cada uno de los puntos y la información de las carreteras a través de las cuales se indica que vamos a realizar la ruta. Finalmente, como suele ocurrir con el uso de este tipo de APIs públicas, también se añade una clave proporcionada desde el portal de desarrolladores de Graphhopper, que permite hacer uso de toda esta funcionalidad.

Así pues, la consulta queda de la siguiente forma:

```
GET https://graphhopper.com/api/1/route?point=39.462160,-0.324177&point=39.441699,-0.595555&vehicle=car&locale=es&elevation=true&instructions=false&points_encoded=false&key=dd2e8e1b-5c27-42e4-8e54-a9f4788fdded&details=max_speed&details=road_class
```

En el caso en el que la consulta se realice correctamente, la respuesta a la misma devuelve la información que se ha pedido en mensaje con formato JSON. En dicho mensaje, se pueden encontrar tres elementos en la raíz del mismo, *hints*, *info* y *paths*, que es el que nos interesa. Dentro de este último elemento, es donde se encuentra disponible la información de la ruta a realizar, junto a una serie de listas.

```
{
  "hints": {...},
  "info": {...},
  "paths": [
    {
      "distance": 34110.374,
      "weight": 2547.098739,
      "time": 2035351,
      "transfers": 0,
      "points_encoded": false,
      "bbox": [-0.612474, 39.436952, -0.324038, 39.481743],
```

La lista *coordinates*, situada dentro del elemento *points* contiene una sucesión de vectores de tres elementos. Cada uno de dichos vectores, hace referencia de uno de los puntos por los que pasa la ruta. Si se viera de forma gráfica, dichos puntos corresponderían a los diferentes puntos que se representan en un mapa para trazar la ruta. Cada uno de estos tres elementos representan la latitud, longitud y altitud de un punto.

```
    "points": {
      "type": "LineString",
      "coordinates": [
        [-0.324179, 39.462164, 1],
        [-0.324289, 39.462153, 0.6],
        ...
        [-0.595565, 39.44166, 147.42]
      ]
    },
    "legs": [],
```

Finalmente, en el apartado *details*, se encuentra la información relacionada a la velocidad máxima permitida por cada tramo y el tipo de carretera a transitar, respectivamente. En el caso del campo *max\_speed*, se trata también de vectores de tres elementos, indicando el identificador del punto de inicio y del punto final donde se aplica la velocidad máxima indicada. Este orden de puntos coincide con el de la lista *coordinates* indicada anteriormente. En el caso de no disponer del valor, un -1 es indicado, tal y como se puede observar en el ejemplo.

En la segunda lista, *road\_class*, se indica de la misma forma que la velocidad (vector de tres valores, con los identificadores de coordenadas) que tipo de carretera se va a transitar. Esto es útil en los casos donde no se dispone de la velocidad máxima, ya que a partir del Reglamento de Circulación<sup>1</sup>, es posible inferir dicha velocidad máxima según el tipo de vía.

```
    "details": {
      "max_speed": [
        [0, 58, -1],
        [58, 60, 50],
```

<sup>1</sup>[http://www.dgt.es/Galerias/seguridad-vial/normativa-legislacion/reglamento-traffic/2015/reglamento\\_trafico184.pdf](http://www.dgt.es/Galerias/seguridad-vial/normativa-legislacion/reglamento-traffic/2015/reglamento_trafico184.pdf)

```
        ...
        [339, 371, -1]
    ],
    "road_class": [
        [0, 8, "residential"],
        [8, 16, "tertiary"],
        ...
        [339, 371, "residential"]
    ]
},
"ascend": 290.8469420671463,
"descend": 144.42344200611115
}
]
```

## 5.2 Modelo de consumo del vehículo

---

Para poder realizar los diferentes cálculos y la simulación de una ruta propuesta, es necesario disponer de un modelo capaz de realizar una estimación lo más aproximada posible del consumo de un vehículo real.



---

---

## CAPÍTULO 6

# Implantación

---



---

---

## CAPÍTULO 7

# Pruebas

---





---

## CAPÍTULO 8

# Conclusión

---

### 8.1 Opciones futuras de líneas de desarrollo

---



# Bibliografía

---

- [1] Benjamin Bader. «An Energy Management Strategy for Plug-in Hybrid Electric Vehicles». En: (nov. de 2013).
- [2] Juan F. Dols. «El Futuro del vehículo eléctrico». En: *Securitas Vialis* 3.3 (oct. de 2011), págs. 67-68. ISSN: 1989-1679. DOI: 10.1007/s12615-012-9046-z. URL: <https://doi.org/10.1007/s12615-012-9046-z>.
- [3] I De Falco, A Della Cioppa y E Tarantino. «Mutation-based genetic algorithm: performance evaluation». En: *Applied Soft Computing* 1.4 (2002), págs. 285-299. ISSN: 1568-4946. DOI: [https://doi.org/10.1016/S1568-4946\(02\)00021-2](https://doi.org/10.1016/S1568-4946(02)00021-2). URL: <http://www.sciencedirect.com/science/article/pii/S1568494602000212>.
- [4] Chiara Fiori, Kyoungcho Ahn y Hesham A. Rakha. «Power-based electric vehicle energy consumption model: Model development and validation». En: *Applied Energy* 168 (2016), págs. 257-268. ISSN: 0306-2619. DOI: <https://doi.org/10.1016/j.apenergy.2016.01.097>. URL: <http://www.sciencedirect.com/science/article/pii/S030626191630085X>.
- [5] T. van Keulen & B. de Jager & A. Serrarens & M. Steinbuch. «Optimal Energy Management in Hybrid Electric Trucks Using Route Information». En: *IFP International Conference* 65 (2010), págs. 103-113. DOI: <https://doi.org/10.2516/ogst/2009026>.
- [6] Lee Daeheung & Suk Won Cha & Rousseau Aymeric & Kim Namwook. «Optimal Control Strategy for PHEVs Using Prediction of Future Driving Schedule». En: *World Electric Vehicle Journal* 5 (mayo de 2012), pág. 149.
- [7] Xinkai Wu y col. «Electric vehicles' energy consumption measurement and estimation». En: *Transportation Research Part D: Transport and Environment* 34 (2015), págs. 52-67. ISSN: 1361-9209. DOI: <https://doi.org/10.1016/j.trd.2014.10.007>. URL: <http://www.sciencedirect.com/science/article/pii/S1361920914001485>.



---

---

## APÉNDICE A

# Glosario

---