



Trabajo de fin de grado

Aplicación móvil para la eficiencia y autonomía de un vehículo eléctrico

Tutora: María de la Peña Fabiani Bendicho

Alumna: Cynthia Alonso Méndez

Grado en Ingeniería Electrónica Industrial y Automática

Curso 17-18

Índice

1. Índice de figuras.....	5
2. Índice de tablas.....	8
3. Glosario	9
4. Objetivo.....	10
5. Abstract.....	11
6. Introducción	12
6.1. Antecedentes	15
7. Alcance y estructura del proyecto	18
8. Componentes de un vehículo eléctrico.....	19
8.1. Batería	19
8.1.1. Características de las baterías	21
8.1.2. Tipos de baterías.....	23
8.2. Motor eléctrico	25
8.2.1. Motores de corriente alterna	26
8.2.1.1. Motores asíncronos.....	26
8.2.1.2. Motores síncronos.....	27
8.2.2. Motores de corriente continua.....	28
8.3. Frenada regenerativa.....	29
8.3.1. Funcionamiento del motor como generador.....	30
8.4. Bloque electrónico de potencia	32
9. Coeficientes para el estudio energético.....	34
9.1. Coeficiente aerodinámico (Cx).....	34

9.2. Coeficiente de resistencia a la rodadura (Crr)	37
10. Estudio energético de un vehículo eléctrico	40
10.1. Pérdidas por energía cinética	40
10.2. Variaciones de energía potencial	41
10.3. Pérdidas por energía aerodinámica.....	41
10.4. Pérdidas por energía de rodadura	42
10.5. Balance energético final	45
11. Descripción de la aplicación: GreenElectricity App	47
11.1. Base de datos	47
11.2. Selección del software: Ionic.....	49
11.3. Lenguajes usados con Ionic.....	51
11.3.1. HTML.....	51
11.3.2. JavaScript (TypeScript)	54
11.3.3. CSS.....	60
11.4. APIs.....	62
11.5. Modos de funcionamiento.....	65
11.6. Funcionamiento de la aplicación	68
11.6.1. Menú a pie de página	69
11.6.2. Menú principal.....	81
12. Posibles mejoras en la aplicación	84
13. Conclusions	86
14. Anexos	87
14.1. Anexo I: base de datos	87
14.2. Anexo II: código de programación de la aplicación	91

1. Índice de figuras

Figura 1- <i>Porcentaje de la UE de emisiones de gases de efecto invernadero</i>	12
Figura 2- <i>Evolución de las ventas desde 2010 a 2017 de vehículos eléctricos e híbridos</i>	13
Figura 3- <i>Ventas de vehículos eléctrico en los últimos años (hasta 2018)</i>	15
Figura 4- <i>Funcionamiento de una batería tanto en carga como en descarga.</i>	20
Figura 5- <i>Campos magnéticos en un motor asíncrono</i>	27
Figura 6- <i>Diferencias entre el motor síncrono y asíncrono</i>	27
Figura 7- <i>Consumo y carga de la batería</i>	30
Figura 8- <i>Cadena de transmisión entre el controlador y las partes del motor</i>	32
Figura 9- <i>Posición del bloque electrónico de potencia dentro del vehículo eléctrico</i>	33
Figura 10- <i>Interior de un túnel de viento</i>	35
Figura 11- <i>Disposición ideal de la forma de un vehículo eléctrico (ovoide)</i>	36
Figura 12- <i>Diagrama de fuerzas en una rueda de un vehículo eléctrico</i>	38
Figura 13- <i>Desplazamiento que tiene que realizar la rueda desde P_1 a P_2 para que ruede sin deslizar</i>	43
Figura 14- <i>Funcionamiento de Ionic</i>	51
Figura 15- <i>Lenguajes que engloba el TypeScript</i>	55
Figura 16- <i>Apariencia del menú horizontal inferior de la aplicación</i>	69
Figura 17- <i>Apariencia de la página de “Inicio” en Android, iOS y Windows respectivamente (parte 1)</i>	69
Figura 18- <i>Apariencia de la página de “Inicio” en Android, iOS y Windows respectivamente (parte 2)</i>	70
Figura 19- <i>Página “Calibrar” (parte 1)</i>	72

Figura 20- <i>Página “Calibrar” (parte 2)</i>	72
Figura 21- <i>Página “Calibrar” (parte 3)</i>	73
Figura 22- <i>Vista de la página del mapa con el marcador inicial y las sugerencias de lugares cercanos del cuadro de búsqueda.</i>	73
Figura 23- <i>Apariencia del botón de “Trazar ruta”</i>	74
Figura 24- <i>Ruta trazada entre el punto actual y el de destino</i>	74
Figura 25- <i>Indicaciones para llegar al destino propuesto y botón de calibrar</i>	75
Figura 26- <i>Cronómetro durante el proceso de calibración.</i>	75
Figura 27- <i>Botones para parar el cronómetro y terminar la calibración</i>	76
Figura 28- <i>Apariencia de la página que aparece al darle a “Funcionamiento normal”.</i>	77
Figura 29- <i>Apariencia de la página donde se calcula la eficiencia en el funcionamiento normal (parte 1).</i>	77
Figura 30- <i>Apariencia de la página donde se calcula la eficiencia en el funcionamiento normal (parte 2).</i>	78
Figura 31- <i>Apariencia de la página de “Quiénes somos” en Android, iOS y Windows respectivamente (parte1)</i>	78
Figura 32- <i>Apariencia de la página de “Quiénes somos” en Android, iOS y Windows respectivamente (parte2)</i>	79
Figura 33- <i>Apariencia de la página de “Quiénes somos” en Android, iOS y Windows respectivamente (parte3)</i>	79
Figura 34- <i>Apariencia de la página de “Contacto” en Android, iOS y Windows respectivamente</i>	80
Figura 35- <i>Apariencia de la página de “Mapa” en Android, iOS y Windows respectivamente (parte1).</i>	80
Figura 36- <i>Apariencia de la página de “Mapa” en Android, iOS y Windows</i>	

<i>respectivamente (parte2)</i>	81
Figura 37- <i>Vista desde el menú en la parte superior derecha desde una de las páginas del menú horizontal</i>	81
Figura 38- <i>Estructura del menú</i>	82
Figura 39- <i>Apariencia de la página que se abre al darle a “iniciar sesión”</i>	82
Figura 40- <i>Apariencia de la página que se abre al darle a “nuevo usuario”</i>	82
Figura 41- <i>Apariencia de la página que se abre al darle a “nueva ruta”</i>	83
Figura 42- <i>Apariencia de la página donde se calcula la eficiencia en el “Funcionamiento normal” (parte 2)</i>	83

2. Índice de tablas

Tabla 1- <i>Etiquetas de HTML usadas en la aplicación</i>	55
Tabla 2- <i>Propiedades de CSS usadas en la aplicación</i>	63
Tabla 3- <i>Datos del Renault ZOE</i>	89
Tabla 4- <i>Datos del Smart ForTwo eD Cabrio, Coupé</i>	89
Tabla 5- <i>Datos del Kangoo Renault ZE</i>	90
Tabla 6- <i>Datos del Nissan LEAF Acenta, Tekna, Visia, Visia+</i>	90
Tabla 7- <i>Datos del Volkswagen e-Golf 100 kW</i>	90
Tabla 8- <i>Datos del Volkswagen e-Golf 85 kW</i>	91
Tabla 9- <i>Datos del BMW i3</i>	91
Tabla 10- <i>Datos del Hyundai IONIQ HE</i>	91
Tabla 11- <i>Datos del Peugeot iON</i>	92
Tabla 12- <i>Datos del Kia SOUL EV</i>	92
Tabla 13- <i>Datos del Tesla Model S 75D</i>	93
Tabla 14- <i>Datos del Tesla Model S P100D</i>	93

3. Glosario

Acrónimo	Significado
UE	Unión Europea
HTML	HyperText Markup Language
JS	JavaScript
POO	Programación orientada a objetos
CSS	Cascading StyleSheets
API	Aplication Programming Interface

4. Objetivo

El objetivo de este proyecto es realizar una aplicación móvil, válida tanto para Android como iOS y Windows, que calcule la autonomía que tendrá el vehículo antes de comenzar, en un trayecto concreto especificado por el usuario. Para ello se hará uso del perfil de alturas que se tiene en dicho trayecto. De esta manera, la autonomía que especifican los fabricantes en las fichas técnicas de los vehículos y se muestra en las aplicaciones de los propios vehículos eléctricos, dejará de ser la real que tendrá el propio vehículo, puesto que esta se calcula en función de la velocidad tomada por el coche en los últimos kilómetros recorridos, en la mayoría de casos. Esta teoría es incierta ya que no contempla la posibilidad de conducción en distintos tipos de vías con diferente velocidad ni tampoco la posibilidad de conducción por el centro de una ciudad o en las afueras, donde existe mucho menos tráfico. Según esta teoría, si un coche ha estado circulando por las afueras la mayoría del tiempo, pero en los últimos kilómetros ha estado dentro de la ciudad, donde el consumo es mucho mayor, indicará al usuario que le queda poca energía almacenada en la batería para conducir, cuando en realidad la batería tendrá mucha más energía acumulada de la que indica puesto que la conducción solo ha sido en ciudad los últimos kilómetros, no todo el tiempo. Es por ello que se considera que estas autonomías proporcionadas por los fabricantes de los vehículos no son del todo fiables, puesto que existen muchos más factores de los que depende la autonomía a parte de la velocidad tomada en los últimos kilómetros.

Este proyecto trata de sustituir este sistema impuesto en los vehículos hasta ahora ya que se tomarán en cuenta las elevaciones del terreno por las que el usuario decida llevar el coche. Por tanto, se obtendrá un valor de autonomía mucho más exacto al no depender de la velocidad de los últimos kilómetros recorridos, teniendo en cuenta las pérdidas ocasionadas por la elevación del terreno y las generaciones de energía en las bajadas, gracias al freno regenerativo.

5. Abstract

This project consists about make an app mobile to know the autonomy (otonomi) of the electric vehicle, given the ground elevation of the journey that the user will travel and its efficiency. This app will made by Ionic, a free framework that works with three languages: HTML, TypeScript and CSS.

The app will have two modes to use it: the first mode is the calibration and it will consist in to do a test with the car to know if the values that the app gives are approximate to the values that the electric vehicle gives; and the second mode is the normal operation, in which it will make an initial hypothesis for to know the kilometres of autonomy that will have the vehicle before to start the travel, so that the user will know to where the vehicle will arrive with the energy that the battery has cumulative (quimulatef).

For do the calculations in the app, it has made an energy study of an electric vehicle calculated all the losses that it is caused by the driving. These losses will add to the caused by the accessories inside of the car and it will obtain the total losses. The result of this energy study is to know how much energy will have to cumulate in the battery for the car finish the journey, in each case with each vehicle.

Also, it has chosen some of the most sent electric vehicle of the Spain market to propose them like an option in the choice in the app, although exists the possibility of choose other that it isn't in the list. It has looking for some of their properties to the energy calculations.

6. Introducción

La obtención de energía más común es a través de recursos fósiles que, aparte de estar limitados, producen grandes niveles de contaminación medioambiental. Este proceso produce emisiones de numerosos compuestos contaminantes que son los responsables de la desertización o del propio cambio climático. La producción actual de energía procede principalmente de dos factores: de la transformación y el consumo de combustibles fósiles, y del consumo de energía (en la industria, los hogares y el transporte, por ejemplo). Esta producción de energía representa el 79 % de las emisiones de gases de efecto invernadero de la UE. Por tanto, la producción de energía y el cambio climático están totalmente ligados actualmente.

Además, España es de los países de la Unión Europea con mayor dependencia energética, teniendo que importar grandes cantidades con respecto al resto. También es uno de los países con más emisiones de gases de efecto invernadero (Figura 1).

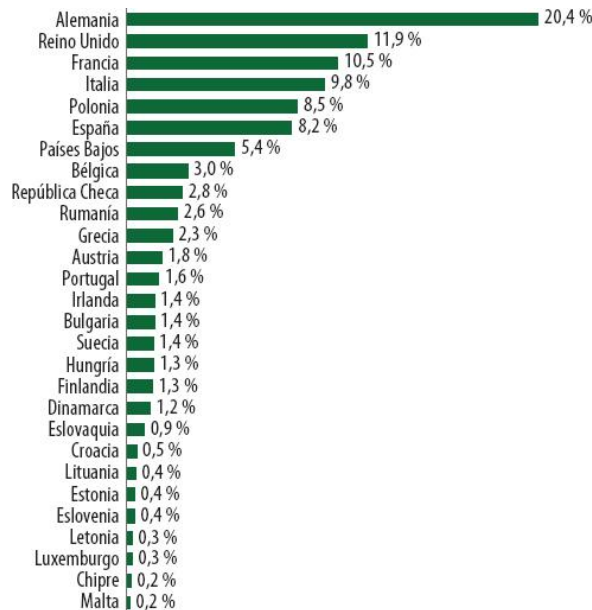


Figura 1: Porcentajes en la UE de emisiones de gases de efecto invernadero. Fuente: EEA greenhouse gas — Data viewer, AEMA, 2017.

Estos son algunos de los hechos que llevan a la sociedad a cambiar el punto de vista y centrarse en un modelo sostenible, eficiente y renovable. Aquí es donde la alternativa de

vehículos eléctricos cobra vida. Se declara como una propuesta limpia, sin emisiones de este tipo de gases, además de muy eficiente y cómoda.

Debido a la conciencia medioambiental que existe, el vehículo eléctrico se convierte en la opción más viable de la actualidad. Las emisiones de los vehículos térmicos ayudan a reafirmar más esta propuesta: a nivel de España, en Madrid y Barcelona se exceden los límites de NO₂ reglamentados por las directivas 2001/81/CE¹ y 2008/50/CE².

Actualmente, la movilidad eléctrica es un tema que está en auge. Esto se comprueba en las cifras de ventas que no paran de subir y en que cada vez se escucha más el vehículo eléctrico como la solución por su vertiente ecológica y de innovación. Aunque hay más de 1,15 millones de vehículos eléctricos en el mundo, solo representa el 0,1% del mercado automovilístico. A pesar de ser un porcentaje bastante bajo, es una cifra que ha ido aumentando exponencialmente en los últimos años, por lo que muestra la aceptación que está teniendo el vehículo eléctrico.

Nos encontramos en plena época de crecimiento del vehículo eléctrico: en los últimos 8 años, las ventas han aumentado notablemente a nivel mundial. En 2010, el número de vehículos eléctricos eran medidos aún en centenas, a diferencia de en la actualidad que, como hemos dicho, pasan el millón de unidades.

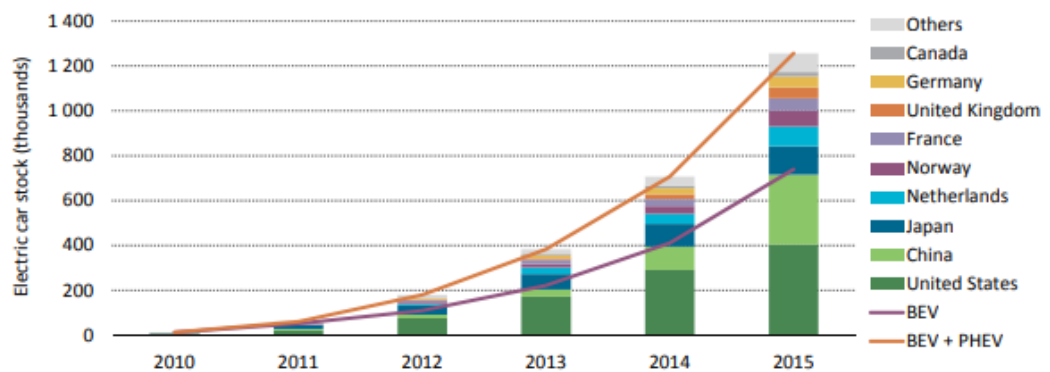


Figura 2: Evolución de las ventas desde 2010 a 2017 de vehículos eléctricos e híbridos.

¹ Directiva 2001/81/EC del Parlamento Europeo y del Consejo, de 23 de octubre de 2001, sobre techos nacionales de emisión para determinados contaminantes atmosféricos.

² Directiva 2008/50/CE del Parlamento Europeo y del Consejo, de 21 de mayo de 2008, relativa a la calidad del aire ambiente y a una atmósfera más limpia en Europa.

Fuente: análisis de la IEA (International Energy Agency) basado en las presentaciones de países de EVI³, complementado por EAFO (2016), IHS Polk (2014), MarkLines (2016), ACEA (2016a), EEA (2015) y IA-HEV (2015).

**Nota: BEV (vehículo eléctrico), BEV + PHEV (vehículo híbrido).*

Según la Agencia Internacional de Energía (IEA), esta mejora se debe a la evolución en las baterías, tanto en su coste como en su autonomía [1]. En los últimos años el coste de la batería se ha dividido por cuatro. Este es un hecho que abarata por tanto el precio final de los vehículos eléctricos, que es uno de los principales problemas de la sociedad para adquirir uno, ayudando así al crecimiento de este tipo de vehículos, a pesar de las prácticamente inexistentes ayudas que existen para poder adquirirlos.

Otro de los factores que influyen a la hora de comprar un vehículo eléctrico es la autonomía. A pesar de los esfuerzos que hacen los fabricantes continuamente para intentar igualar la autonomía a la de un vehículo de combustible, siguen sin ser equiparables. Además, la infraestructura de estaciones de recarga sigue siendo escasa [2], lo que hace que se cree una dependencia de la recarga del vehículo desde casa y además que solamente se puedan utilizar para trayectos cortos. Una de las ventajas que se propone en la aplicación de este proyecto es el conocimiento de la autonomía con más exactitud, según el perfil de alturas, lo que podría ayudar al usuario ya que este podría calcular exactamente las paradas necesarias para las recargas que tendría que realizar.

Como muestra de ello, el mes de enero de 2018 ha mantenido los niveles de ventas de coches eléctricos de los meses finales del año 2017 y se convierte en el mes de enero con mayor número de matriculaciones de la historia superando al mismo mes del año 2017 en un 226%. Precisamente la ciudad de Madrid es la que encabeza el número de matriculaciones de vehículos eléctricos. La diferencia con los últimos años se muestra en la siguiente figura:

³ Electric Vehicle Initiative.

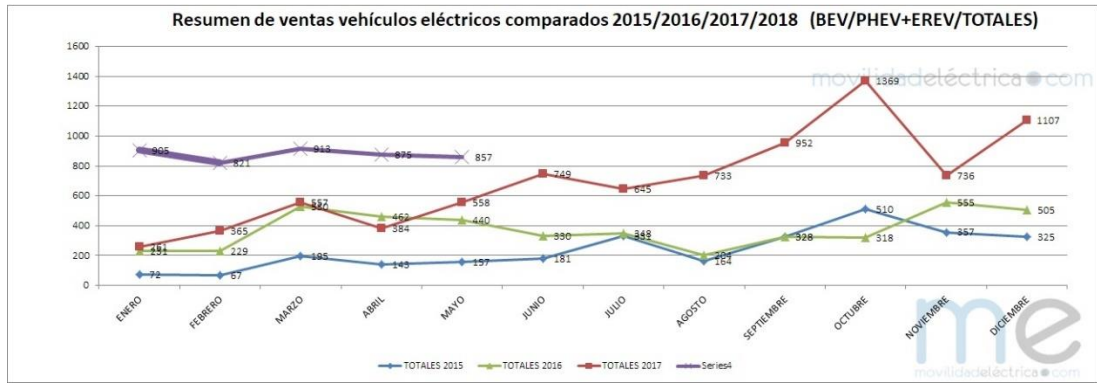


Figura 3: Ventas de vehículos eléctrico en los últimos años (hasta 2018).

Dada la situación de evolución a nivel mundial y concretamente en España sobre los vehículos eléctricos, este proyecto se enfoca como un estudio actual e innovador, atrayente para todo aquel que posea un vehículo de estas características. Incide, además, como ya hemos comentado, en uno de los puntos más influyentes en la opinión de los compradores de este tipo de vehículos: el control de la autonomía.

Por otro lado, las tecnologías se presentan como una manera de vida prácticamente hoy en día. En una época en la que es imposible pensar vivir sin teléfono móvil, esta aplicación que se ha creado permite facilitar la vida del usuario del vehículo de manera que se le aportan datos mucho más exactos a través de este dispositivo. Se trata de una manera rápida y fácil de llegar hasta el usuario.

6.1. Antecedentes

El vehículo eléctrico fue uno de los primeros automóviles que se desarrolló en el siglo XIX, antes de la llegada de los motores de combustión.

El primer vehículo eléctrico constaba de pilas que no eran recargables, por lo que no era una máquina con mucha utilidad. Posteriormente aparecieron las primeras baterías recargables y fue cuando el coche eléctrico tomó más importancia ya que contaban con un gran rendimiento. Incluso a principios del siglo XX fueron los más que se vendieron, aunque hay que tener en cuenta que la alternativa, que eran los motores de combustión, no eran como lo son ahora, sino que tenían muchas más desventajas que ventajas, aparte de

ser incómodos para la clase social que disponía de un automóvil. Pero los vehículos eléctricos cayeron en el olvido debido a la llegada del motor de arranque de combustión. El precio de la gasolina mucho más asequible y la construcción de mejores carreteras ayudaron a que se diera este acontecimiento.

La crisis del petróleo y la conciencia ecológica retomaron del olvido estos vehículos, aunque sin mucho éxito. Fue a principios del siglo XXI cuando vuelven con fuerza los automóviles totalmente eléctricos. Se implantan en nuestras vidas con un rendimiento del 90% frente al 38% de los vehículos de motor de combustión, con un menor coste de combustible (energía) y con un mayor confort.

Esta llegada responde a la necesidad de reducir la dependencia del petróleo y con ello las emisiones de gases de efecto invernadero causantes del cambio climático, abriendo paso a una nueva conciencia y manera de pensar de la sociedad.

Tanto los vehículos eléctricos como los de combustible, cuentan con un sistema de predicción de autonomía que, o bien se muestra en el cuadro o cuentan con un navegador que lo muestra, una aplicación móvil o un sistema para integrar las aplicaciones a través de una pantalla táctil. En concreto, prácticamente todos los vehículos eléctricos dan la opción al usuario para descargar una aplicación móvil en la que se muestran todos los parámetros funcionales del vehículo, como la intensidad que está siendo consumida o los kilómetros que le quedan a la batería disponibles por consumir. Actualmente, en cualquiera de las alternativas nombradas para mostrar los parámetros, se muestra la autonomía calculada según la velocidad tomada en los últimos kilómetros, una forma que hace que se devuelva un valor aproximado, pero no todo lo real que se podría. Aquí aparece la mejora que plantea nuestra aplicación y que la diferencia de estas que ya existen: calculará los kilómetros disponibles en función del perfil de alturas que se vaya a recorrer, de manera que se conocerán las pérdidas cuando el vehículo esté subiendo, la generación de energía cuando esté bajando y las pocas pérdidas que consumiría cuando vaya en llano (idealmente, en llano el vehículo ni consume ni genera energía).

Nuestra aplicación objeto de este proyecto, podrá proporcionar un valor de autonomía mucho más exacto y dotará de seguridad al usuario al saber que el dato que se está proporcionando es lo más real posible. Además, como ya se ha nombrado en este mismo apartado, el usuario podrá planificar las paradas que tendrá que realizar para recargar la batería con más precisión que si lo hiciera con la aplicación del propio vehículo. Desde la

aplicación podrá saber su posición actual y concretar la ruta que va a recorrer, conociendo instantáneamente las indicaciones que tendrá que seguir para llegar a su destino, el tiempo aproximado y los kilómetros totales de su trayecto. Al hacer uso de la Google Maps API, el tiempo aproximado estará actualizado constantemente y será lo más real posible puesto que tiene en cuenta el tráfico y cualquier posible alteración en las vías que se vayan a tomar en el trayecto.

Algunas de las aplicaciones para vehículos eléctricos que existen son:

- Green Charge: una aplicación que muestra el rango en el que se encuentra la intensidad consumida por la batería para saber cuándo recargarla, permitiendo al usuario conocer su forma de conducción y mejorarla para su menor consumo.
- Green Race: llevada a cabo por un grupo de estudiantes y profesores, se trata de una aplicación que muestra el tanto por ciento de batería restante que le quedaría al terminar un trayecto, utilizando el perfil de alturas. Es bastante similar a la aplicación de este proyecto.

Como hemos dicho, también existen las propias de los vehículos eléctricos, como NissanConnect EV o Z.E. Services (Renault), e integrada en el propio cuadro del coche como en el caso del Kona (Hyundai).

7. Alcance y estructura del proyecto

El alcance del proyecto consta de un estudio energético de un trayecto que realiza un vehículo eléctrico, de un estudio de los principales vehículos del mercado automovilístico y de la creación de una aplicación móvil manejando un *framework* desconocido hasta ahora que es Ionic.

Para conseguir este objetivo, ha sido necesario adquirir conocimientos esenciales como el estudio de los componentes que forman parte del vehículo que han sido de interés para el trabajo, conocimientos acerca de las características más importantes de un automóvil eléctrico y el aprendizaje de tres nuevos lenguajes como son HTML, TypeScript y Sass.

En cuanto a la estructura, primero se explicarán los componentes más importantes del vehículo eléctrico, los distintos tipos que hay y cuáles son los que se utilizan en los vehículos objetos de este proyecto. Además, se explicarán en profundidad las características de dos coeficientes de los que se hace uso en el estudio energético.

Luego, se mostrará el estudio de mercado que se ha hecho para seleccionar los vehículos eléctricos más vendidos en la actualidad y las características que han sido útiles.

El siguiente apartado será el estudio energético. Se mostrarán las fórmulas que se han utilizado y se explicarán los distintos tipos de pérdidas/recuperación de energía que sufre el vehículo a medida que avanza.

Por último, se explicarán los lenguajes y el *framework*⁴ utilizados, y la estructura de la aplicación creada, con su funcionamiento y distintos modos de uso.

Como anexo se mostrará todo el código de programación que se ha utilizado en los tres lenguajes nombrados anteriormente.

⁴ *Framework*: es un entorno de trabajo compuesto por una estructura conceptual y tecnológica con módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

8. Componentes de un vehículo eléctrico.

Aunque este proyecto trata sobre un estudio energético y la creación de una aplicación móvil, debemos conocer el objeto sobre el que vamos a trabajar, que en este caso es el vehículo eléctrico, y su composición. Es por ello que se explicarán a continuación cuáles son los componentes esenciales de este y cómo funcionan.

Un vehículo eléctrico está compuesto principalmente por el motor eléctrico, la/s batería/s y el bloque de potencia, que es el que conecta el motor y la batería y hace posible que estos puedan trabajar. No está compuesto solo por estos tres, sino que existen muchos más componentes, aunque no son de gran relevancia en este proyecto. Por tanto, sólo se hablará de aquellos que sean importantes a la hora de crear la aplicación y que hay que tenerlos en cuenta. Es decir, se dejarán atrás las explicaciones de elementos como las propias ruedas, las características físicas del chasis (sí se hablará de su forma y dimensiones) o el circuito eléctrico interno.

8.1. Batería

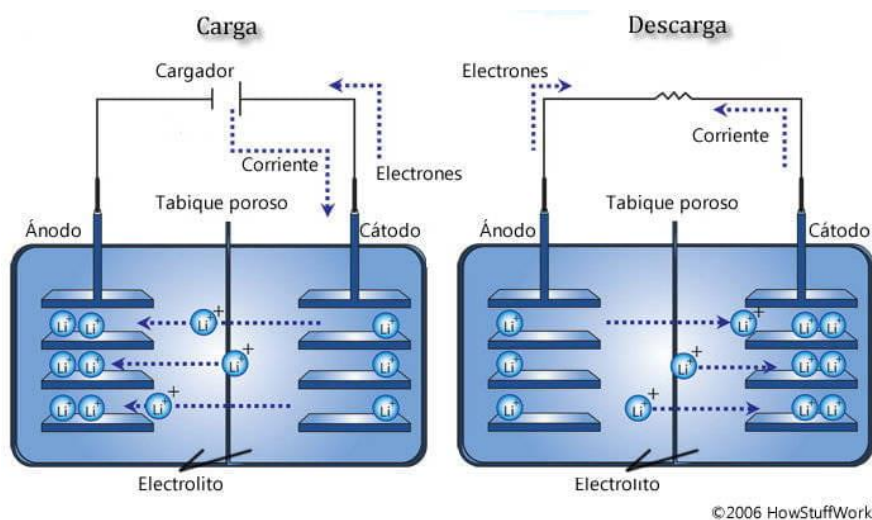
Uno de los componentes más importantes en el interior de un vehículo eléctrico son las baterías. La autonomía que tendrá el vehículo dependerá de esta, de su capacidad y de su energía almacenada; de ahí su gran importancia. Por tanto, se trata del elemento equivalente al depósito de combustible en un vehículo tradicional. Su función es almacenar energía y transformarla en energía eléctrica para enviarla al motor.

El interior de estas consta de un conjunto de celdas, compuestas a su vez por dos electrodos y un electrolito, y en cada una de ellas es donde se produce la reacción química que permite la creación de energía eléctrica, reacción que consiste en el flujo de iones y electrones entre los dos electrodos. De estos electrodos, uno es positivo (cátodo) y otro es negativo (ánodo), y son los que permiten que la corriente pueda salir de la batería hacia el motor.

Los electrodos son los extremos de un conductor que están en contacto con un medio, en este caso el electrolito, que es la sustancia en la que se encuentran los iones y hacen que

se comporte como conductor eléctrico. Aquí es donde se produce la reacción química de oxidación-reducción (redox), que es reversible, ya que los productos que la componen vuelven a combinarse para generar reactivos, y se ceden o se absorben electrones según se esté cargando o descargando. Por ello, se genera una tensión eléctrica entre los dos electrodos que, como están conectados con el exterior, también se genera por tanto en los bornes de la celda.

Este proceso químico se comporta distinto en dos casos: cuando el coche está cargándose y cuando está descargándose. Cuando está en modo de descarga, el proceso de redox consiste en el traspaso de electrones del electrodo positivo al ion, generándose así una sustancia con exceso de electrones (reducción en el positivo) y crea una sustancia con déficit de electrones debido del traspaso de electrones del electrodo negativo al ion (oxidación en el negativo). A medida que se va cargando la batería, estas sustancias se van acumulando en los polos, creando un desequilibrio electrónico que, al conectar el motor con la batería, produce energía eléctrica. En cambio, en el proceso de carga se invierten: se reduce el negativo y se oxida el positivo, permitiendo la recuperación de los electrones perdidos.



© 2006 HowStuffWorks

Figura 4: Funcionamiento de una batería tanto en carga como en descarga.

Fuente: “Global autos: don’t believe the hype-analyzing the cost & potential of fuel-efficient technology – Bernstein Research & Ricardo”.

8.1.1. Características de las baterías

- Las baterías recargables no tienen una vida indefinida, sino que van perdiendo capacidad con su uso. Es decir, soportan un número determinado de ciclos de carga y descarga y es lo que se conoce como ciclo de vida. Esto ocurre debido a un producto químico residual que se crea en cada recarga de la batería que se va acumulando y va restando espacio a los iones operativos. Aun así, se trata de un factor que está en estudio y su evolución es notable en las baterías de los últimos modelos de vehículos eléctricos.

A parte, las baterías tienen un cierto margen de energía acumulada que nunca gastarán y que, si el usuario decide forzar la batería y consumirla, podría romperla. Se trata de la profundidad de descarga y afecta a la vida útil de la batería, pues si el usuario está constantemente forzando la batería, esta disminuirá.

- Uno de los elementos que se tendrán en cuenta en el objeto de este proyecto es la eficiencia de carga. Las baterías no consumen todo lo que cargan; es decir, existen pérdidas que hacen que la eficiencia no sea del 100%. Estas pérdidas son las debidas a las reacciones químicas que se producen en el interior de las celdas nombradas anteriormente. Por ello, la energía que realmente entrega no es toda la que se introduce. En nuestro caso, cuando no se tengan datos concretos del modelo de batería utilizado, se tomará una eficiencia del 80%. Esto quiere decir que, del total de energía que se carga en la batería, el 80% se destinará al movimiento del vehículo y el resto se perderá en forma de calor. Se escoge este dato porque las baterías que se usarán serán de ion-Litio [3].

Mirándolo desde el punto de vista del balance energético de estudio, una vez se tienen todas las pérdidas de energía ocasionadas por el movimiento del vehículo, la pregunta que nos hacemos es cuánto tiene que cargar la batería para poder consumir todas las pérdidas ocasionadas, teniendo en cuenta las pérdidas debidas a las reacciones químicas. No podemos cargar lo mismo que consumimos (este sería el caso ideal), sino que tiene que ser más para poder contemplar dichas pérdidas que se transforman en calor (caso real).

- Otra característica que influirá en la autonomía del vehículo es la temperatura. Esta influye en la reacción química que se produce en el interior de las celdas ya que el calor aumenta la facilidad del cambio de electrones, por lo que la reacción química sucederá más rápido. Por ello, la temperatura ambiente es la conveniente y cuentan con un sistema de refrigeración para intentar mantenerla constante. Si no lo consiguen, las temperaturas altas y sobre todo las bajas, afectan negativamente en la disminución de los kilómetros de autonomía de la batería. El rango de temperatura para obtener el mayor rendimiento está entre 5 y 30 °C, por lo que España es un país idóneo para el uso de las baterías de ion-Litio de los vehículos eléctricos que se usan en este proyecto (concretamente, Canarias aún más).

Según un estudio realizado por ARC de AAA⁵, a temperaturas por encima de la temperatura ambiente, la autonomía puede disminuir un 34,7% [4]. Aunque a altas temperaturas se favorece la carga de la batería, también favorece otras reacciones como la corrosión que, al no ser convenientes, se instala un sistema de refrigeración como ya hemos dicho para evitarlas. En cambio, cuando se trata de temperaturas bajas, la autonomía se ve afectada en una disminución del 59,3%. Este número indica que se perderán unos 20 km por cada decremento de 5 grados. Esta cifra aumentaría con el uso de la calefacción unos 10 km más, por lo que supondría la pérdida de 30 km de autonomía cada 5 grados.

Algunas de las soluciones que se han implantado en modelos actuales de vehículos eléctricos para sustentar los cambios térmicos y evitar el uso de la calefacción son, por ejemplo, el preacondicionamiento del vehículo mientras se está cargando o el precalentamiento de la batería en condiciones frías.

- La autonomía que tendrá el vehículo hemos dicho que será la que tenga la propia batería. Entendemos por autonomía como la cantidad de energía eléctrica que se acumula transformada en distancia. Esto nos dará un número de kilómetros que será capaz de recorrer el vehículo.

⁵ ARC de AAA: "Automotive Research Center" de "American Automobile Association". Se trata de un centro de investigación automotriz perteneciente a la asociación americana de automóviles.

La energía acumulada será, por tanto:

$$E_{acum} = Capacidad * V [Wh]$$

Esta capacidad es la capacidad de carga y es la que puede almacenar la batería. Se da en amperios-hora (Ah). Al multiplicarlo por el voltaje al que se somete, se obtiene la energía acumulada en la batería que nos dará su autonomía. En la capacidad de los últimos vehículos eléctricos del mercado, ya se está teniendo en cuenta la profundidad de descarga óptima, es decir, el tanto por ciento de batería que no debe descargarse, puesto que disminuye su vida útil.

- La densidad energética es la cantidad de energía que puede entregar la batería por cada kilogramo de su peso. Se trata de la relación entre la autonomía del vehículo y el peso de la propia batería. Una densidad energética alta equivale a que la batería será ligera y además podrá entregar mucha batería, que es lo más conveniente. Esta característica está relacionada con el peso de la batería, por lo que afecta directamente al peso del vehículo, que es uno de los parámetros más importantes en el estudio energético de este proyecto. Las baterías pueden llegar a pesar hasta 300 kg, por lo que no se puede despreciar su efecto en el peso total del vehículo. Por ello, la densidad energética es una característica importante a la hora de elegir la batería a usar y conviene saber de qué se trata puesto que las baterías propias del estudio de este trabajo tienen una buena densidad energética. Se explica esta característica de las baterías para poder entender posteriormente una de las razones del uso de las baterías de ion-Litio en los vehículos eléctricos.

8.1.2. Tipos de baterías

Existen varios tipos de baterías según los materiales y elementos químicos utilizados que se nombrarán a continuación, ya que el único tipo que se explicará con más detalle serán las baterías de ion-litio, debido a que son las que se utilizan en los vehículos utilizados para este proyecto.

- Baterías de plomo y ácido: electrolito de ácido sulfúrico y electrodos de plomo. Se trata de baterías que han quedado desplazadas por su baja densidad y eficiencia.
- Baterías de Níquel y metal: dentro de esta categoría se encuentran las baterías de Níquel-Hierro, Níquel-Cadmio y Níquel-Hidruro metálico. Tienen una potencia buena, un ciclo de vida largo y no presentan problemas medioambientales. Están por encima de las de plomo y ácido, pero lejos aún de las de ion-litio.
- Baterías de ion-Litio: es el tipo de batería más extendida en los coches eléctricos puesto que poseen pequeño volumen y peso, y alta potencia y densidad energética. Utilizan un electrodo positivo (cátodo) de óxido de cobalto, trifilina u óxido de magnesio y un electrodo negativo (ánodo) de grafito. Su desarrollo es más reciente y están en constantes investigaciones para su mejora. Como ya se nombró antes, estas baterías no toleran la descarga completa, por lo que llevan una circuitería adicional para conocer el estado de la batería. En el caso de la recarga, pueden hacerlo sin necesidad de estar descargadas completamente y, en este caso, no se reduce su vida útil.

Como desventaja cabe destacar que no soportan bien los cambios de temperaturas ni de voltaje. Por ello, incorpora un sistema de gestión de la batería que se encarga de realizar funciones para que el uso de la batería se encuentre dentro de unos valores que se consideran seguros y controla el correcto funcionamiento de esta. Este sistema es bastante costoso y, junto a la propia composición química, hace que se aumente el coste del vehículo cuando se incorpora respecto al resto de baterías.

A pesar de sus inconvenientes, se trata de baterías que, por su voltaje, densidad energética, eficiencia de recarga y ciclo de vida, son muy superiores a los otros tipos de baterías.

Aunque se tratan como un único tipo de baterías, estas engloban varias ramas como son las de Litio-Cobalto, Litio-Hierro-Fosfato, Litio-Manganeso, Litio-Níquel-Cobalto-Manganeso y Litio-Titanio.

Estas son las baterías más utilizadas, con claro dominio de las de ion-Litio, que están en continuo estudio y posiblemente se mejoren aún más sus características con el paso del tiempo. El diseño de nuevos tipos de baterías es un hecho y pronto se planteará la posibilidad de incorporarlas (baterías con nanotecnología, supercondensadores, baterías de litio en estado sólido, baterías de metal-aire).

Una de las baterías que pronto se implantarán y de la que se le ha concedido la patente a Tesla son las de metal-aire como carga. La idea sería utilizar las baterías de ion-Litio para desplazamientos cortos y para los desplazamientos largos, la batería de metal-aire recargando a la de ion-Litio.

8.2. Motor eléctrico

Un motor eléctrico es una máquina rotativa que convierte la energía eléctrica en energía mecánica de rotación. Es por ello que también se dice que es una máquina motriz. Consta de una parte fija (estator) y una parte móvil (rotor). En su interior se producen las fuerzas de giro que impulsan el vehículo y hace que se mueva. Además, todos los motores poseen un rendimiento y tienen pérdidas debidas a los mecanismos, al hierro y al cobre. Cuando las pérdidas ocasionadas por el cobre son iguales al resto de pérdidas, el motor trabajará a pleno rendimiento. Existen motores de corriente continua y corriente alterna y dentro de los de alterna se diferencian los motores síncronos y asíncronos.

Este proyecto se centrará en los vehículos eléctricos. En ellos, el movimiento se produce por interacciones electromagnéticas a diferencia de los vehículos de motor en los que la energía se obtiene por medio de explosión.

Algunos motores pueden ser reversibles; es decir, pueden convertir también la energía mecánica en eléctrica como es nuestro caso. Cuando funciona de esta manera, se dice que

el motor funciona como generador. Se da cuando el vehículo se desacelera o se frena en el caso de los vehículos eléctricos gracias al freno regenerativo (**apartado 8.3.**).

Según el tipo de máquina eléctrica, su funcionamiento será uno u otro. En nuestro caso, los vehículos que se han estudiado para realizar este proyecto, todos trabajan con motores de corriente alterna.

8.2.1. Motores de corriente alterna

En este tipo de motores, en el estator se encuentran las bobinas, que son elementos conductores, y están inmersas en un campo magnético. Al hacer pasar una corriente por dichas bobinas, se crea un campo magnético contrario al ya existente que hace que se mueva la parte móvil del motor: el rotor. Este principio es el de inducción y es en el que se basa el funcionamiento del motor.

8.2.1.1. Motores asíncronos

El motor asíncrono está formado por el rotor, que según su disposición puede ser de jaula de ardilla o de bobinado, y el estator, que es donde se encuentran las bobinas inductoras trifásicas. Su funcionamiento es el siguiente: al aplicar corriente alterna trifásica en el estator, se crea un campo magnético que envuelve al rotor y produce una corriente eléctrica que desemboca en la creación de una fuerza que lo hace girar. Esta fuerza es la causante del segundo campo magnético que gira en el mismo sentido que el primero, pero no a la misma velocidad (aquí se produce el deslizamiento, que es la diferencia entre ambos campos). Estas dos velocidades nunca serán iguales puesto que cuanto más se acerque la velocidad a la del primer campo, más pequeña será la intensidad inducida.

Este tipo de motores presentan las ventajas de alta eficiencia, bajo coste, produce poco ruido y vibraciones, y par constante. Por otro lado, cuentan con las desventajas de baja densidad de potencia, bajo par de arranque y poseen riesgo de sobrecarga.

Tanto el Tesla S 75 D como el Tesla S P100D, vehículos objeto de estudio en este proyecto, cuentan con un motor de este tipo, trifásico y con rotor bobinado.

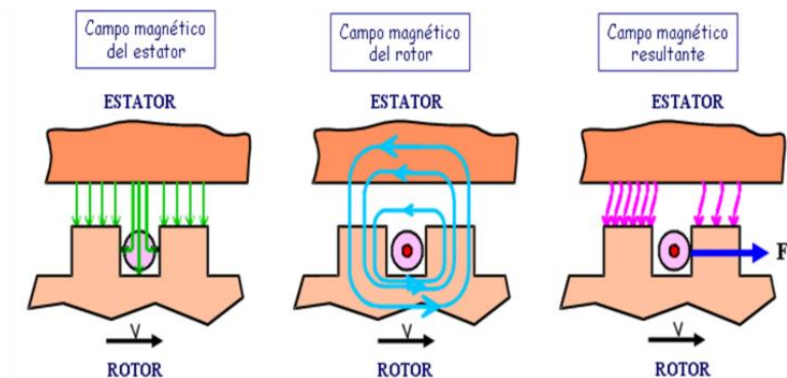


Figura 5: Campos magnéticos en un motor asíncrono.

8.2.1.2. Motores síncronos

Un motor síncrono es aquel en el que su velocidad es proporcional a la frecuencia de la corriente que circula por él. En este caso, el inductor es el rotor y el inducido es el estator y el campo magnético que se crea en el estator y el del rotor, giran a la misma velocidad exactamente (velocidad sincrónica, que es la velocidad del campo magnético rotante que se crea en el estator) al contrario que en las máquinas asíncronas, y se produce un par. Es por ello que en este tipo de máquinas no existe el concepto de deslizamiento y además se puede conseguir la potencia que se quiera.

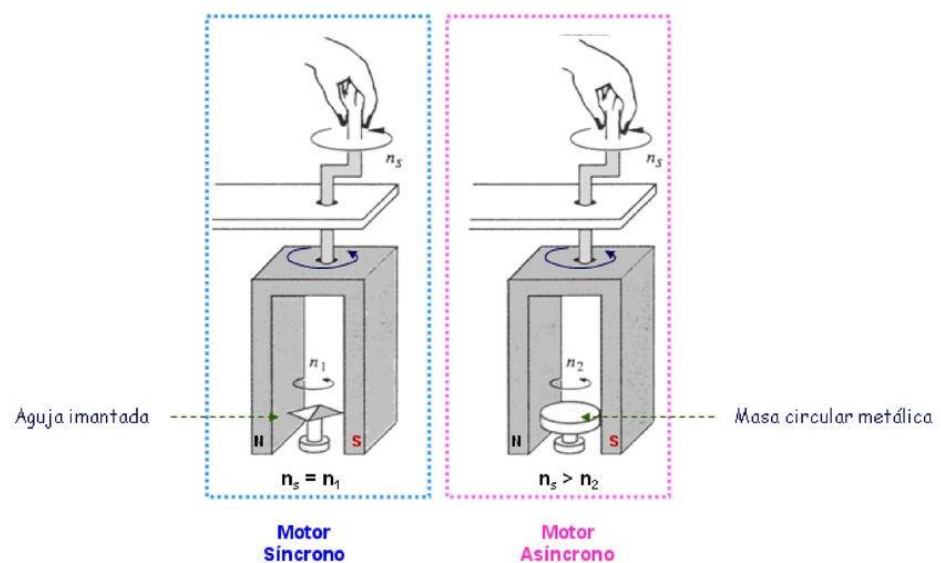


Figura 6: Diferencias entre el motor síncrono y asíncrono.

Mediante las fuerzas de atracción y repulsión de los polos, el rotor no gira, sino que vibra. Existen distintos modos de arranque según el modelo de vehículo, que hacen girar inicialmente a la velocidad de sincronismo y hasta que se produce un enganche magnético que hace que el rotor siga girando sin necesidad ya de esta medida auxiliar de arranque.

Estos motores proporcionan un par elevado a una velocidad constante bajo cargas variables, lo que se traduce en costes de mantenimiento y explotación reducidos. Debido al mayor rendimiento, menor tamaño y mayor relación de potencia de salida, estos motores pueden sustituir a los motores de corriente continua. Además, tienen un mejor rendimiento y precisión que los asíncronos, pero, comparados con estos, los motores síncronos tienen una construcción más compleja y por tanto son menos económicos.

Existen dos tipos de motores síncronos: de imanes permanentes y de reluctancia conmutada. Los de imanes permanentes constan de 3 electroimanes en el rotor y los de reluctancia conmutada de un objeto de hierro que se alinea en el tiempo con el campo magnético externo.

Los motores síncronos de imanes permanentes son los que utilizan el Mitsubishi iMIEV, Peugeot iON, Nissan LEAF Acenta, Tekna, Visia y Visia+, BMW i3, Renault ZE, Renault ZOE, Hyundai IONIQ HE y el Smart ForTwo eD Cabrio y Coupé, que son vehículos que se estudian en este trabajo. Como vemos, la mayoría de los vehículos eléctricos estudiados, utilizan este tipo de motor puesto que tienen un alto rendimiento y costes de mantenimiento bajos, a pesar de su elevado precio.

Por otro lado, en los motores síncronos de reluctancia conmutada o variable, la corriente es conmutada entre las bobinas del estator, creando un campo magnético que hace que gire el rotor. No necesitan ni imanes permanentes ni escobillas para funcionar. No existe ningún vehículo estudiado en este proyecto con este tipo de motor.

8.2.2. Motores de corriente continua

Existen también los motores de corriente continua. Consiste en transformar la corriente alterna inducida en una espira, al girar dentro de un campo magnético, en

corriente continua. Funcionan igual que una máquina de corriente alterna síncrona, pero en este caso se transformará la corriente alterna de la señal, presente en el inducido, en una señal continua (unidimensional) en el circuito exterior, a través del colector de delgas. Una de las diferencias entre los motores de continua y los de alterna es que el par de arranque es mucho mayor en los de continua, tienen velocidades variables y la velocidad se puede ajustar aumentando la tensión. En cambio, los de alterna operan a velocidades fijas y no pueden funcionar a bajas velocidades.

Los motores de corriente continua siempre han sido más útiles que el resto debido a su mayor grado de flexibilidad para el control de la velocidad y del par, pero han sido sustituidos por los motores de alterna por estos tener un coste de mantenimiento y fabricación menores. En la década de los 90, todos los vehículos eléctricos funcionaban con motores de corriente continua.

La implantación de los motores de corriente alterna en los vehículos eléctricos en lugar de los motores de corriente continua se debe a que en los de alterna, las pérdidas no crecen con el tamaño de la máquina, permitiendo un alto rendimiento; a que el precio es mucho mayor en los de continua, debido a los imanes permanentes y, además, los de alterna son mucho más fáciles de proteger debido a que no producen tensión cuando se les deja de suministrar corriente. A pesar de que los motores de corriente alterna tienen mayores costes de desarrollo, a largo plazo supondrán menores gastos.

Por estos motivos, los motores de corriente continua se verán más en vehículos híbridos y los de corriente alterna en los vehículos totalmente eléctricos.

8.3. Frenada regenerativa

El freno regenerativo es un sistema con el que cuentan los vehículos eléctricos para recuperar las pérdidas de energía producidas mediante la frenada o desaceleración en la conducción. Consiste en aprovechar la energía cinética del frenado convirtiéndola en energía eléctrica y almacenándola para una posterior reutilización en la batería. Este sistema caracteriza a los vehículos eléctricos frente a los vehículos con motores térmicos

que al frenar trabajan por rozamiento y expulsan toda esa energía hacia el exterior en forma de calor.

Como es de esperar, no todo lo que se consume es recuperado mediante este sistema, sino que existe un rendimiento, ya que ningún sistema de recuperación de energía es perfecto. En el caso de este proyecto, cuando no se conozcan datos concretos del modelo de coche estudiado, se tomará una eficiencia de carga de un 80%. Representa que recupera el 80% de lo que pierde durante el frenado [5]. Esta eficiencia de carga influirá en la autonomía del vehículo por lo que cuanto mayor sea mejor.

En el caso de los vehículos eléctricos, la energía de frenado se utiliza para recargar las baterías actuando igual que un generador. Pero no son necesarias dos máquinas independientes como un motor y un generador, sino que un motor puede funcionar como generador invirtiendo su funcionamiento. Como ya hemos dicho, es el caso de algunos motores que se hacen llamar reversibles debido a esta característica.

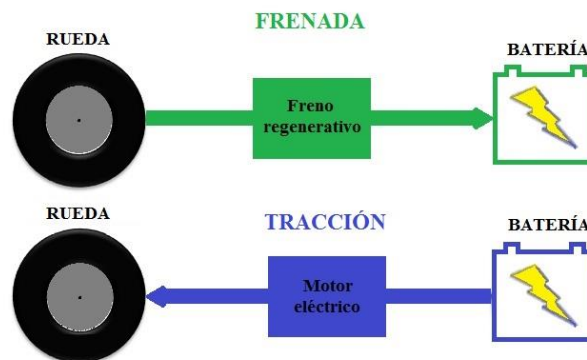


Figura 7: Consumo y carga de la batería.

8.3.1. Funcionamiento del motor como generador

Los elementos que intervienen en este proceso son los más importantes de un coche eléctrico: el controlador eléctrico, el motor y las baterías. El controlador eléctrico es un componente que regula la cantidad de energía que pasa al motor y comprueba su correcto funcionamiento.

El motor eléctrico consta principalmente de dos partes: el rotor (móvil) y el estator (inmóvil), como ya hemos dicho (**apartado 8.2.: Motor eléctrico**). Se hablará de motores síncronos pues utilizaremos su versatilidad para funcionar tanto como motor como generador. Para que funcione como motor, se ha de introducir corriente alterna en el estator. Cuando esto ocurre, se genera un campo magnético que se induce en el rotor y hace que este se mueva. Por tanto, el funcionamiento del coche cuando se está acelerando, es decir, cuando el motor funciona como tal, es el siguiente: el controlador capta corriente continua de las baterías y la convierte en alterna para introducirla en el estator. Esta corriente es la que crea el campo magnético en el rotor provocando que este se mueva ocasionando el movimiento del coche. El campo magnético que se crea dependerá de la intensidad magnética, la cual será mayor cuanto más se acelere el coche por medio del usuario mediante el pedal del acelerador.

Cuando el motor funciona como generador es porque se desacelera, es decir, se pone en marcha el freno regenerativo. Esto es: el controlador deja de suministrar la corriente al estator por lo que deja de existir el campo magnético que producía el movimiento. En este caso, a quien envía corriente es al rotor que al moverse induce un campo magnético que a su vez induce una corriente eléctrica alterna en las bobinas del estator. Esta corriente alterna llega al controlador que es el que se encarga de transformarla en continua para almacenarla en las baterías. Esto solo es posible gracias a la fuerza de frenado que se ejerce en el vehículo debido a la desaceleración. Esta fuerza será mayor si el frenado aumenta, por lo que aumentará si además se pisa el pedal del freno. Cuanto más se frene, más intensidad introducirá el controlador en el rotor, más intensidad se generará y mayor será el campo creado. En el momento que se pise bastante el freno, entran en acción además los frenos mecánicos del vehículo ya que el frenado regenerativo no es un sistema que se instale solo, sino que necesita la ayuda también las pastillas y discos de frenos. El uso combinado de ambos se traduce en unos menores costes de mantenimiento comparado con los coches de combustión.

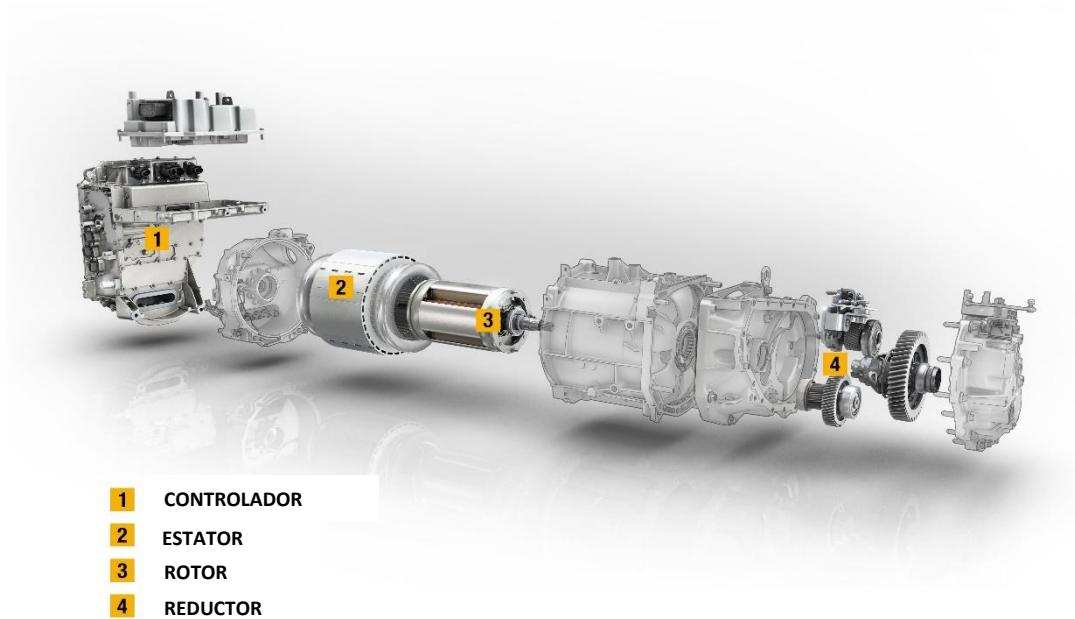


Figura 8: cadena de transmisión entre el controlador y las partes del motor.

Fuente: Página web oficial de Renault.

8.4. Bloque electrónico de potencia

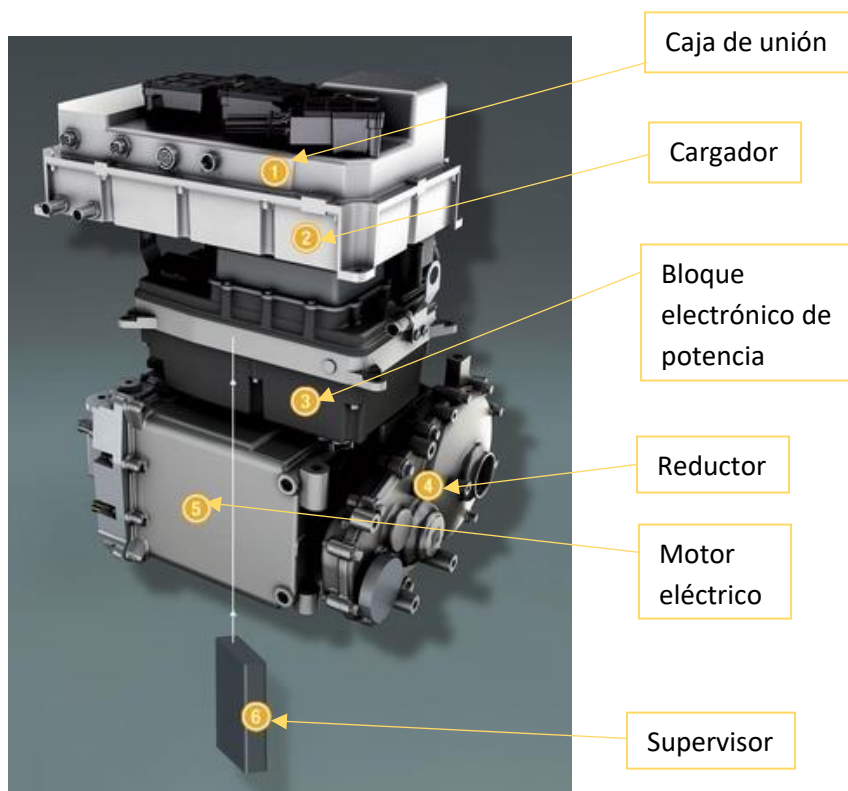
El controlador del que se habla en el freno regenerativo (**apartado 8.3.**), es un elemento que se encuentra dentro de uno de los sistemas más importantes del coche eléctrico: el regulador eléctrico o bloque electrónico de potencia. Está compuesto a su vez por un inversor, un rectificador, un transformador y el controlador:

- El inversor es el encargado de convertir la corriente continua procedente de las baterías en corriente alterna.
- El transformador se ocupa de equilibrar los voltajes entre las baterías y el motor ya que trabajan a 200 V y 600 V respectivamente. Hace por tanto de intermediario para equilibrar estos voltajes.
- El rectificador tiene la función contraria al inversor, convierte la corriente alterna en continua para poder almacenarla en las baterías cuando el motor esté funcionando como generador.
- Por último, el controlador organiza todos los elementos descritos que componen el regulador según las especificaciones del conductor. Es el encargado de absorber la

energía de las baterías para transportarla hasta el motor. Así, al presionar el acelerador, se activan las resistencias encargadas de transmitir la señal al controlador y este calcula la energía necesaria que hay que enviar para que el motor funcione. Es el elemento que interpreta las indicaciones del conductor, respecto a la aceleración o freno.

Se hace llamar el director del sistema energético del vehículo puesto que, junto con las baterías y el motor, forman a grandes rasgos los componentes esenciales para que el vehículo se mueva y sin él, ni el motor ni las baterías podrían funcionar.

Toda la energía que entra o sale del motor pasa necesariamente por el regulador. Por ello, este sistema debe ser eficiente y ocasionar pocas pérdidas ya que su eficiencia afectará directamente a la eficiencia global del vehículo y, por tanto, a su autonomía.



9:

bloque electrónico de potencia dentro del vehículo eléctrico.

*Figura
posición del*

Fuente: Página oficial de Renault: Twizy ZE.

9. Coeficientes para el estudio energético

A continuación, se detallarán dos de los coeficientes que influyen en los cálculos del estudio energético que se explicará más adelante. Estos coeficientes se detallan por su gran importancia en la autonomía del vehículo y en las pérdidas de este.

9.1. Coeficiente aerodinámico (C_x)

El coeficiente aerodinámico representa la forma aerodinámica que tiene el vehículo. Al multiplicarlo por la superficie, se obtiene la resistencia que opone al paso por el aire debido a su forma. Es un concepto que está ligado a la eficacia y eficiencia del vehículo. Conviene que este coeficiente sea pequeño puesto que cuanto más pequeño sea, se tendrá una forma del vehículo más aerodinámica y facilitará su movimiento.

Este coeficiente se puede obtener de manera práctica, a través de túneles de viento y maquetas a escala, haciendo uso de las técnicas de análisis dimensional, que consisten en conseguir un conjunto mínimo de números adimensionales a partir de los parámetros dimensionales de entrada, simplificando así el estudio y el número de pruebas a realizar para determinar el coeficiente.

Los túneles de viento están compuestos por un ventilador gigante que crea ráfagas de viento muy grandes, equipados con unas rejillas estabilizadoras al comienzo para garantizar el tipo de flujo que se necesite enviar según las pruebas a realizar. Se coloca en la zona de prueba la maqueta a escala del vehículo y se colocan sensores de presión en su superficie. Posteriormente, se simula el movimiento del vehículo a través del aire ya que se sitúa dicha maqueta en una superficie móvil (el vehículo permanece estático pero las ruedas no). De estos ensayos se obtienen distribuciones de presión y valores de las fuerzas que actúan, que son las que nos permitirán mejorar las propiedades aerodinámicas del vehículo, como el C_x .



Figura 10: interior de un túnel de viento con un vehículo eléctrico. En él se le somete a pruebas para ajustar su coeficiente aerodinámico.

También se puede obtener de manera teórica a través de la dinámica de fluidos computacional, que trata de resolver las ecuaciones de la mecánica de fluidos aplicadas al vehículo.

En el caso del coeficiente aerodinámico, este se verá reflejado en el cálculo de las pérdidas de energía debidas a la aerodinámica, ya que es un factor indispensable para poder calcularlas.

Existen todo tipo de disposiciones, pero la forma ideal es un ovoide (como un huevo) o una cuña porque, según José Ramón Sánchez, jefe de Laboratorio del Departamento de Aerodinámica de INTA (“Instituto Nacional de Técnica Aeroespacial”):

“Su frontal es una curva suave que penetra fácilmente en el aire y su parte trasera alargada devuelve suavemente el aire a su posición original, evitando las temidas turbulencias”.

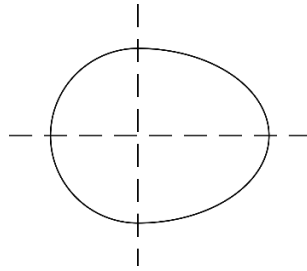


Figura 11: disposición ideal de la forma de un vehículo eléctrico (ovoide).

Debería ser la forma de gota la mejor forma para un vehículo ya que es la más aerodinámica que existe puesto que ofrece la mejor relación entre volumen y resistencia. Esto no es posible llevarlo expresamente a la práctica porque conllevaría una pérdida de confort en pasajeros distribuyendo los asientos de otra manera a la que estamos acostumbrados. Aun así, los fabricantes intentan asemejar la forma lo más posible a esta.

El C_x , como ya hemos dicho, no representa la resistencia aerodinámica que tiene que vencer el vehículo, sino que lo hace cuando se multiplica por la superficie frontal. Por ello, el coeficiente aerodinámico por sí solo no depende de la superficie del coche. Es por esto que, un vehículo grande con un bajo coeficiente aerodinámico puede terminar teniendo mayores pérdidas aerodinámicas que uno de menor tamaño con un coeficiente más alto.

El coeficiente aerodinámico puede presentarse de igual manera en dos coches totalmente distintos. Un vehículo puede tener un área frontal muy grande y tener un coeficiente aerodinámico muy pequeño debido a su forma que es muy aerodinámica, y otro tener un área frontal más pequeña pero su forma no ayuda a la aerodinámica por lo que tendrá un coeficiente aerodinámico mayor. Ambos vehículos tendrán el mismo rozamiento, por lo que vemos que este coeficiente no depende solo del área frontal sino también de su forma.

Actualmente es un factor que está en continuo estudio para conseguir disminuirlo. Los fabricantes han de combatir el viento puesto que condiciona el diseño, la estabilidad, produce ruidos e influye en el consumo del vehículo. Además, los vehículos también necesitan aire en el interior y refrigerar aquellos elementos que desprenden calor como lo es el radiador, por ejemplo. Este factor también está en estudio para su mejora ya que lo que se busca es que el vehículo pueda estar refrigerado sin que suponga la disminución de velocidad del coche y sin tener que aumentar su C_x ; es decir, que no tenga prácticamente impacto sobre este.

Puesto que disminuir el C_x por medio de la forma del coche supondría realizar estructuras muy poco confortables y seguras, muy poco llamativas para el consumidor, hoy en día, lo que los fabricantes buscan es disminuirlo por medio de los parachoques, aletas, parabrisas y las tomas de aire, buscando así la forma más aerodinámica posible.

9.2. Coeficiente de resistencia a la rodadura (C_{rr})

El coeficiente de resistencia a la rodadura indica el gasto de energía por calor debido a la rodadura del vehículo. La resistencia a la rodadura es la segunda fuerza más importante que se opone al avance del vehículo después de la resistencia aerodinámica, aunque a bajas velocidades, la resistencia a la rodadura supone una carga incluso mayor que la resistencia aerodinámica.

Para que este coeficiente tenga sentido, hemos de tener un cuerpo deformable y otro indeformable, que en nuestro caso se trata de las ruedas y el asfalto, respectivamente. Las deformaciones que se producen se dan gracias a las presiones en los puntos de contacto entre ambas superficies, aunque también se puede generar por el rozamiento interno de los rodamientos de las ruedas, aunque en mucha menor medida.

Para ver la fuerza que se tendrá que vencer para que el cuerpo ruede, y, por tanto, conocer la energía de la rodadura, se hará un balance energético en una rueda individual que se extiende a las ruedas restantes:

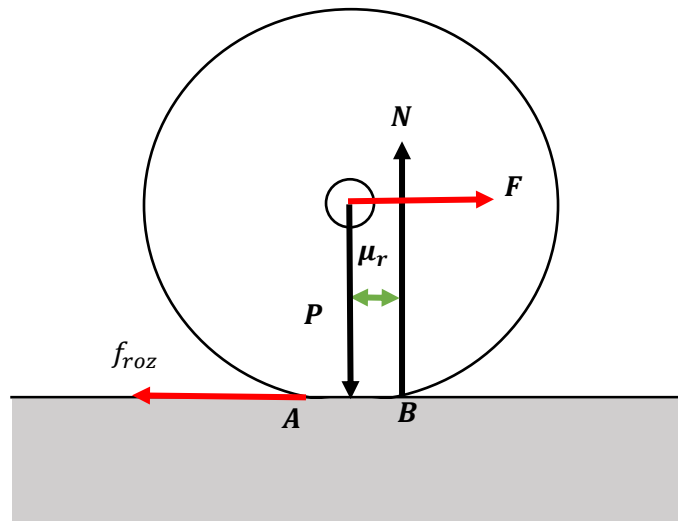


Figura 12: Diagrama de fuerzas en una rueda de un vehículo eléctrico.

- Como podemos observar, las fuerzas que actúan son la fuerza normal N que ejerce la superficie sobre la rueda, el peso del vehículo P , la fuerza F que provoca el movimiento y por ello, existe también la fuerza de rozamiento f_{roz} que se opone.
- El movimiento se generará a lo largo de toda la superficie de contacto con el pavimento; es decir, desde A hasta B en la figura.
- Con el objetivo de simplificar los cálculos, se considera que la fuerza que hay que vencer será la normal N , que no tendrá su línea de acción sobre la del peso, sino desplazada una distancia μ_r para poder rotar y superar el obstáculo de la existencia de la deformación de la rueda. Por ello, se creará un par que será el de resistencia a la rodadura, a parte del par del motor.

$$M_{motor} = R \cdot F$$

$$M_{rodadura} = \mu_r \cdot N$$

Donde R es el radio de la rueda.

- Como hemos dicho, habrá que superar dicho par que se crea para que la rueda pueda girar. Esto equivale a que el par del motor sea mayor al par que se crea por la rodadura:

$$M_{motor} \geq M_{rodadura} \Rightarrow R \cdot F \geq \mu_r \cdot N$$

- Despejando, obtenemos la fuerza F que habrá que aplicar para que la rueda gire sin deslizar:

$$F \geq \frac{\mu_r \cdot N}{R} = C_{rr} \cdot N$$

- Como vemos, cuando mayor sea el radio de la rueda, menor es la fuerza que hay que ejercer para que se mueva; es decir, R y F son inversamente proporcionales. Además, deducimos la ecuación del coeficiente de rodadura como:

$$C_{rr} = \frac{\mu_r}{R}$$

- En general, este coeficiente suele ser menor que el coeficiente de rozamiento. Por ello, en los vehículos se sustituyen, en la medida de lo posible, los deslizamientos que provoca la fuerza de rozamiento por rodaduras, disminuyendo así las pérdidas energéticas. Por este motivo, en nuestro cálculo de energías, se han obviado las pérdidas por rozamiento, sin dejar de tener en cuenta que sí influyen en otros aspectos como la eficiencia global.

Esta fuerza F es la que se contemplará en el cálculo de las pérdidas de energía debido a la rodadura. Es por ello que es tan importante el conocimiento del coeficiente de rodadura.

10. Estudio energético de un vehículo eléctrico

El objetivo de este trabajo es conocer la autonomía que tendrá el vehículo a priori; es decir, antes de que el conductor inicie el trayecto ya sabrá hasta dónde le permitirá conducir este con la energía que tiene cargada en la batería. Dicha autonomía es la cantidad de energía eléctrica que se acumula en la batería transformada en distancia. Por tanto, para calcular cuánta será la energía que necesita para realizar el trayecto marcado, será necesario conocer todas las pérdidas energéticas que tendrá a medida que el vehículo avanza. Equivale a conocer cuánta energía habrá que almacenar en la batería para terminar el trayecto fijado o hasta dónde llegará con la que ya tiene. Esto es, realizar el cálculo energético que se procede a explicar a continuación.

Las variaciones energéticas (pérdidas y ganancias) que se producen a medida que el coche avanza serán las debidas a la energía cinética, potencial, aerodinámica y de rodadura.

10.1. Pérdidas por energía cinética

Se trata de la energía más evidente que influye en el consumo del coche dado que es la energía asociada al movimiento; es decir, se trata de la cantidad de trabajo que se consume a medida que este avanza. Es por ello que, para conocerla, hemos de conocer parámetros como la masa del vehículo, sumada a la masa de los pasajeros a bordo, y la velocidad a la que se viaja, obteniendo las pérdidas asociadas a esta energía mediante la siguiente fórmula:

$$E_c = -\frac{1}{2} \cdot m \cdot v^2$$

m = masa del vehículo + masa de los pasajeros (kg)

v = velocidad a la que circula el vehículo (m/s)

10.2. Variaciones de energía potencial

Como ya se ha nombrado a lo largo del texto (**apartado 6.1.**), se tomará en cuenta el perfil de alturas que se va a recorrer. Es decir, el vehículo gastará más energía cuando sube que cuando circula en llano y obtendrá energía en bajada mediante el freno regenerativo. Así, cuando el vehículo esté subiendo por una carretera, la energía potencial será negativa y, por el contrario, cuando baje será positiva. Es por ello que esta energía tendrá gran importancia y para hallarla se tendrán que conocer parámetros como la pendiente del trayecto, por ejemplo. Debido a la influencia del freno regenerativo en este tipo de energía, se tiene en cuenta un factor que llamamos “factor de eficiencia regenerativa”. Sabemos que toda la energía que se consume al subir, no se genera totalmente al bajar, sino que será un poco menos del 100%. Esto es lo que representa dicho factor. Así, por ejemplo, si su valor es de un 80% (0.8), se consumirán 10 kWh en una subida y, mediante el freno regenerativo, se podrán recuperar 8 kWh.

$$E_p = -m \cdot g \cdot h$$

m = masa del vehículo + masa de los pasajeros (kg)

g = constante de gravitación universal (m/s²)

h = altura del tramo (m)

10.3. Pérdidas por energía aerodinámica

Se trata de la energía que pierde el vehículo debido a la resistencia que opone a medida que avanza, inmerso en el aire, debido a su forma. Es por ello que para que la influencia de esta energía sea menor, se tendrá que tener en cuenta la forma aerodinámica del coche, que se refleja en el área frontal ya que el flujo se separa una vez se topa con esta superficie. Cuanto más aerodinámica sea, mayor velocidad podrá alcanzar el vehículo y las pérdidas por energía aerodinámica serán menores.

Para representar cuán efectiva es la forma del coche, existe el “coeficiente aerodinámico”, que se representa como C_x (adimensional) ya que se relaciona con el eje longitudinal del vehículo (área frontal). Al multiplicarlo por la superficie frontal, es cuando se obtiene la resistencia que se opone al paso por el aire. Representa

expresamente la facilidad del vehículo para penetrar en el aire. Cuanto menor sea, menor resistencia se opone y es más conveniente para el vehículo por lo comentado anteriormente. Cada vehículo tiene su coeficiente aerodinámico, es constante y lo suele proporcionar el fabricante.

En el caso de los vehículos de carretera, este coeficiente depende básicamente de su propio diseño. El C_x se obtiene a partir del conjunto del parabrisas, espejos laterales y ponente, y la parte baja del frente del coche. Disminuirán este factor las formas suaves de los faros, la defensa y el resto de obstáculos que por lo general dificultan la aerodinámica del coche.

Por tanto, para conocer estas pérdidas de energía será necesario saber parámetros como la superficie frontal, el coeficiente aerodinámico y la distancia del trayecto a recorrer:

$$E_{aerod} = -\frac{1}{2} \cdot C_x \cdot v^2 \cdot A \cdot d$$

$C_x =$ coeficiente aerodinámico (adimensional)

$v =$ velocidad $\left(\frac{m}{s}\right)$

$A =$ superficie frontal del vehículo (m^2)

$d =$ distancia total a recorrer

10.4. Pérdidas por energía de rodadura

Se trata de la energía que se pierde cuando el vehículo rueda sin deslizar. Esto es, que el punto de apoyo de las ruedas no se separa con respecto al suelo.

La rodadura es un movimiento que combina la traslación con la rotación y se asocia al movimiento de las ruedas.

“Para que la esfera ruede sin deslizar, el desplazamiento del CM (centro de masas) debe coincidir con el arco s correspondiente al ángulo girado, según se aprecia”⁶:

⁶ Fuente: Universidad Politécnica de Madrid (Montes).

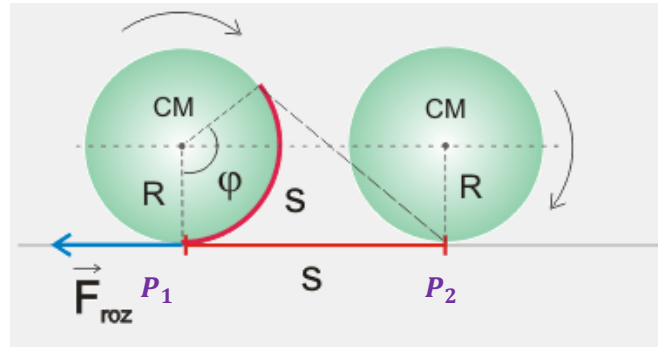


Figura 13: desplazamiento que tiene que realizar la rueda desde P_1 a P_2 para que ruede sin deslizar.

A medida que el vehículo avanza, los neumáticos se deforman ante la superficie por la que circula que es indeformable, calentándose y consumiendo energía en forma de calor, provocando así la existencia del “coeficiente de resistencia a la rodadura (C_{rr})” presente en todos los neumáticos. Cuanto mayor sea este coeficiente, mayor energía se gastará mediante el contacto con la superficie (conviene que sea pequeño). Este coeficiente representa expresamente la energía que se consume en forma de calor debido al contacto de dos superficies, una indeformable y otra deformable.

Por tanto, para conocer este tipo de energía, se deberán conocer parámetros como el C_{rr} , la distancia a recorrer y el peso del vehículo sumado al peso de los pasajeros:

$$E_{rodadura} = -C_{rr} \cdot m \cdot g \cdot d$$

C_{rr} = coeficiente de resistencia a la rodadura (adimensional)

g = constante de gravitación universal ($\frac{m}{s^2}$)

m = masa del vehículo + masa de los pasajeros (kg)

d = distancia recorrida

La fuerza que aparece en este tipo de movimiento (F_{roz}) es la fuerza de rozamiento y es aquella que se opone al desplazamiento de un cuerpo cuando este está en movimiento debido al contacto entre dos superficies. Es por ello que, para que el cuerpo se mueva, habrá que aplicarle una fuerza mayor para que pueda vencer la fuerza de rozamiento, que tiene sentido opuesto al movimiento.

Esta es independiente de la velocidad, solo depende de la fuerza normal, que es la que se opone al peso del cuerpo y es perpendicular al desplazamiento, y del coeficiente de rozamiento.

“El coeficiente de rozamiento o coeficiente de fricción vincula la oposición al deslizamiento que ofrecen las superficies de dos cuerpos en contacto según la intensidad del apoyo mutuo que experimentan”⁷.

Este coeficiente depende de la naturaleza de los materiales de las superficies y es adimensional. Cuanto menos rugosas sean las superficies en contacto, menor será este coeficiente. En cambio, si se trata de dos superficies que oponen mayor resistencia al deslizamiento, tendrá un coeficiente de rozamiento cerca de la unidad o incluso mayor. Por tanto, el objetivo es que se trate de un coeficiente de rozamiento bajo.

Se pueden diferenciar dos tipos de fuerzas de rozamiento: aquella fuerza que se opone al movimiento (fuerza de rozamiento dinámica), y aquella que existe antes de que el vehículo se mueva y esta se opone a que lo haga (fuerza de rozamiento estática). Generalmente, la fuerza estática es mayor que la dinámica.

Este concepto de rozamiento se tiene en consideración en el trabajo que se expone a la hora de definir la eficiencia global del vehículo. La fuerza de rozamiento no se tiene en cuenta a la hora de calcular las energías ya que para simplificar los cálculos se omite. Esta simplificación se ha visto en el apartado de “Coeficiente de resistencia a la rodadura” (**apartado 9.2.**). Es por esto que no entra en la fórmula de las pérdidas debidas a la rodadura. En cambio, sí que se tiene en cuenta aplicando una eficiencia global del 80% y no del 100%, ya que tiene bastante influencia en el movimiento del vehículo oponiéndose a este, aunque no entre específicamente en el cálculo de ninguna energía.

Una vez se tienen las pérdidas y ganancias que se realizan en cada intervalo de distancia fijado, se suman todas y se obtiene finalmente el balance energético del trayecto.

⁷ Fuente: Wikipedia.

10.5. Balance energético final

Aparte de estas energías provocadas por el movimiento del propio vehículo eléctrico, existen unas pérdidas que hay que contemplar que son las debidas a los componentes internos del coche. El bloque electrónico de potencia es el sistema que une el motor con la batería para que el vehículo pueda arrancar. Es por tanto un elemento muy importante que tiene como tarea transformar la electricidad que proporciona la batería en electricidad de menor voltaje para el motor. Esto provoca que se genere calor, que se manifiesta en forma de pérdida de energía y son las pérdidas que llamamos debidas a los accesorios. En nuestro caso, se toma una potencia de 1.5 kW en los accesorios que, dividido por la suma de los tiempos empleados en el trayecto, se obtienen las pérdidas de energía que ocasionan estos componentes internos del vehículo. [6]

Para conocer entonces la energía total consumida por el vehículo al final del trayecto, primero se sumarán todas las pérdidas de energía nombradas anteriormente y se le hará la operación correspondiente con la eficiencia global. A esta suma no se les añaden las pérdidas debidas a los accesorios ya que a este tipo de pérdidas no les afecta el rendimiento, ya que, como hemos dicho, se trata de pérdidas internas del vehículo y no dependen de la conducción del coche. Por tanto, esta suma de pérdidas será la energía útil y lo que nosotros queremos obtener es la energía consumida: la energía total que se consume durante todo el recorrido del vehículo, desde que sale hasta que se para. En el caso ideal, ambas energías serán iguales, pero en nuestro caso consideraremos una eficiencia inferior al 100% debido a factores no contemplados en los cálculos de pérdidas de energía anteriormente descritos, como por ejemplo la fuerza de rozamiento. Es por ello que, para calcular la energía consumida, hemos de tener en cuenta que esta será siempre mayor que la útil y, por tanto:

$$E_{consumida} = -\frac{E_{útil}}{\mu} [kWh]$$

Una vez obtenido el balance energético es cuando se les suman las pérdidas debidas a los accesorios anteriormente descritas, y obtendríamos la energía total consumida.

Analizando el resultado obtenido para un trayecto determinado, la batería tendrá que proporcionar los kWh consumidos en forma de pérdidas o bien se recargará con los

producidos en forma de generación. El problema es que la batería también tiene una eficiencia y no entrega todo lo que tiene acumulado, sino que tiene pérdidas. Es decir, si queremos que la energía acumulada en la batería sea como mínimo la consumida en el trayecto para así poder terminarlo, debemos cargar un número de kWh mayor para que al tener en cuenta su eficiencia, pueda conseguir igualar el número de la energía perdida. Para tener en cuenta este efecto, se introduce en la programación, en las fórmulas del balance energético, la división de la suma de todas las pérdidas y ganancias (energía útil) entre el rendimiento del vehículo.

11. Descripción de la aplicación: GreenElectricity App

Hoy en día, todos los vehículos tienen un sistema que determina los kilómetros que le quedan hasta gastar el depósito. En el caso de los vehículos de combustión, se muestra en el cuadro del coche o en cualquier otra parte del vehículo, los kilómetros que faltan para que se vacíe el depósito de combustible. En el caso de los vehículos eléctricos, todos ellos están dotados de un sistema, generalmente apto para teléfonos móviles, el cual muestra características eléctricas del motor, como por ejemplo la intensidad que se está consumiendo, y también, entre otras, la autonomía del vehículo; es decir, los kilómetros que le quedan hasta que la batería se descargue. El problema de este sistema es que calcula dicha autonomía en función del consumo durante los últimos kilómetros. Esto quiere decir que no contempla la posibilidad de que el vehículo vaya más rápido de lo que ha ido hasta ahora o de que comience un trayecto que es en continua subida, por ejemplo. En este ámbito se sitúa este trabajo de fin de grado, ya que se trata de una aplicación móvil para vehículos eléctricos que calcula la autonomía, pero dependiendo del trayecto que se vaya a recorrer, de manera que antes de iniciar el recorrido, el usuario sabrá si su vehículo está cargado suficientemente para realizar todo el trayecto o si de lo contrario se quedará a mitad de este ya que la batería no está suficientemente cargada. Así, por ejemplo, dependiendo del perfil de alturas que se extraiga del trayecto a recorrer, el consumo será mayor o menor, teniendo en cuenta el freno regenerativo que se pondrá en uso en las bajadas que se tomen, o en función de la velocidad media de la zona que se va a recorrer (urbana o autopista) se consumirá más o menos electricidad en el trayecto.

Tras explicar el funcionamiento de los componentes más importantes del vehículo y el cálculo energético que se ha realizado, a continuación, se detallará el proceso que se ha seguido para crear la aplicación móvil. Además, se explicarán los lenguajes de programación que se han utilizado y el *framework*.

11.1. Base de datos

Para la aplicación de este proyecto, se ha hecho un previo estudio de los vehículos eléctricos más importantes del mercado en la actualidad. Tras observar las estadísticas y

algunos estudios⁸, se han escogido un total de 12 modelos distintos de vehículos con diferentes características, siendo las opciones más atractivas hoy en día y las más vendidas. En esta lista está incluido también el mejor modelo eléctrico de uso industrial (Renault ZE) y dos modelos de Tesla S, no por ser los más vendidos sino por ser el mejor vehículo eléctrico del mercado, con la mayor autonomía y la mayor rapidez para alcanzar los 100 km/h.

Las características de todos estos modelos se han incluido en la aplicación como base de datos de partida. Naturalmente, el uso de dicha base de datos es opcional, pudiendo utilizarse para cualquier otro modelo siempre que se introduzcan los parámetros adecuados.

Una vez escogidos los modelos de estudio según el mercado en España, se ha enfocado el interés en una serie de características como lo son la autonomía o la capacidad de la batería entre otras, descritas en los apartados anteriores. Además, ha sido vital obtener de cada modelo el ancho, el alto, el peso y el coeficiente aerodinámico para poder obtener la superficie frontal y calcular los distintos tipos de pérdidas y ganancias durante el trayecto.

Como se verá a continuación, la mayoría cuenta con motores síncronos de imanes permanentes y todos tienen baterías de ion-litio; por ello se explican sus funcionamientos en este proyecto.

La capacidad de la batería y la autonomía se incluyen para saber cuánto puede acumular el coche y para conocer la estimación inicial de la que se parte con respecto a la autonomía, antes de poner en práctica la aplicación. Son datos interesantes en este proyecto puesto que se hará una predicción de las pérdidas y ganancias que tendrá el vehículo en el trayecto marcado antes de iniciarlo. Al conocer la capacidad de la batería, se podrá saber hasta qué punto del trayecto llegará el coche con la batería que tiene acumulada y cuándo tendrá que recargarlo.

En el apartado de Anexos (**apartado 14.1.**), se pueden observar una serie de tablas con los datos que se han indicado de cada modelo que se ha escogido.

⁸ Fuentes: El País y www.movilidadeléctrica.com.

11.2. Selección del software: Ionic

Ante la idea de hacer una aplicación móvil, existen muchas opciones para llevarla a cabo, que se dividen en gratuitas y de pago, y en crear la aplicación móvil programando o sin programar. Las opciones que no necesitan de programación suelen ir ligadas a que sean de pago. Como ejemplo de estas se encuentran: Just In Mind, Mobincube, AdianteApps, App Press o Apps Builder.

Por otro lado, se encuentran las opciones que incluyen programación, como por ejemplo Appcelerator Titanium, PhoneGap o jQuery Mobile.

Para este proyecto se ha escogido una opción que permita programación y sea gratuita. Se trata del software Ionic, una plataforma que utiliza tres lenguajes (HTML, JavaScript y Sass) de manera sencilla e intuitiva: su codificación se basa en desarrollo web. Permite crear la aplicación para cualquier sistema operativo, por lo que se tratará de una aplicación híbrida. También permite visualizar los cambios a medida que se crea a través de la web. Dispone de tres modos de utilizarlo: Ionic Framework, Pro y Enterprise. En este proyecto se ha utilizado el básico (Ionic Framework) puesto que las características que poseen las otras dos opciones son de pago, poseen un nivel de opciones superior al empleado y, por ello, no se utilizarían en este caso.

A excepción de la primera versión de Ionic en la que se trabajaba con JavaScript, el resto de versiones (incluida la que se utiliza en este proyecto) trabaja con TypeScript. Este tipo de lenguaje es igual que JavaScript, pero con mejoras entre las que destaca la incorporación del uso de clases, por lo que se hace un lenguaje de POO, con código más legible y con mayor control sobre él. Este tipo de lenguaje hace que sea sostenible para grandes proyectos.

Su mayor ventaja es que se puede compilar con un solo código en todas las plataformas (iOS, Android y Windows), ahorrando tiempo y recursos.

Ionic crea aplicaciones recopilando lo mejor de las aplicaciones web y de las aplicaciones nativas. Las aplicaciones web hacen uso de un “web view”, que se trata de un navegador que ejecuta código HTML, JavaScript y CSS, con lo que se podrá construir una página web que se convertirá en una aplicación. Las aplicaciones híbridas, como las

que crea Ionic, toma un “*web view*”⁹, pero en lugar de ejecutarlas directamente en el navegador, tal y como hacen las aplicaciones web, se empaqueta junto con todos los “*assets*”¹⁰ en un formato u otro en forma de código nativo¹¹, dependiendo de la plataforma en la que se vaya a utilizar. Por tanto, una aplicación híbrida consiste en mostrar una página web dentro del contenedor de una aplicación móvil con toda la lógica y base de datos que necesita para funcionar de forma autónoma.

Ionic está basado en Cordova. Se trata del software que hace de nexo entre el *web view* donde se ejecuta el código y la comunicación con el dispositivo, mediante código nativo. Además, provee unas APIs¹² escritas en JavaScript (plugins) que permiten interactuar con las características del dispositivo, como lo son la cámara, el correo, la geolocalización, etc. Este código que se ejecuta en el *web view* es el que se crea mediante Ionic y se comunica con Cordova para acceder a las APIs del dispositivo.

El motor de Ionic es Angular. Se trata de un *framework* de desarrollo creado por Google cuya finalidad es la ejecución de aplicaciones web. Es quien procesa las peticiones HTTP, se conectará a servicios, etc. Permitirá la creación de una web, que posteriormente será una aplicación, que será dinámica y tendrá una comunicación con el servidor de acción-reacción, en donde las peticiones del usuario se harán prácticamente de forma instantánea. De esta forma no se tendrá que cargar las páginas consultadas, sino que se hará todo sobre una única página.

En la siguiente figura se muestra un esquema del funcionamiento de Ionic y el procedimiento que sigue hasta llegar a la pantalla de la aplicación:

⁹ Vista de la aplicación por medio del navegador web.

¹⁰ Se trata de las hojas de estilo CSS y los archivos JavaScript que hacen que la aplicación tenga un buen aspecto.

¹¹ Código o aplicaciones nativas: se trata de escribir en el lenguaje propio del sistema operativo en el que se vaya a ejecutar la aplicación.

¹² APIs (*Application Programming Interfaces*): Conjunto de comandos, funciones y protocolos informáticos que permiten crear programas específicos para ciertos sistemas operativos. Permite la comunicación entre softwares.

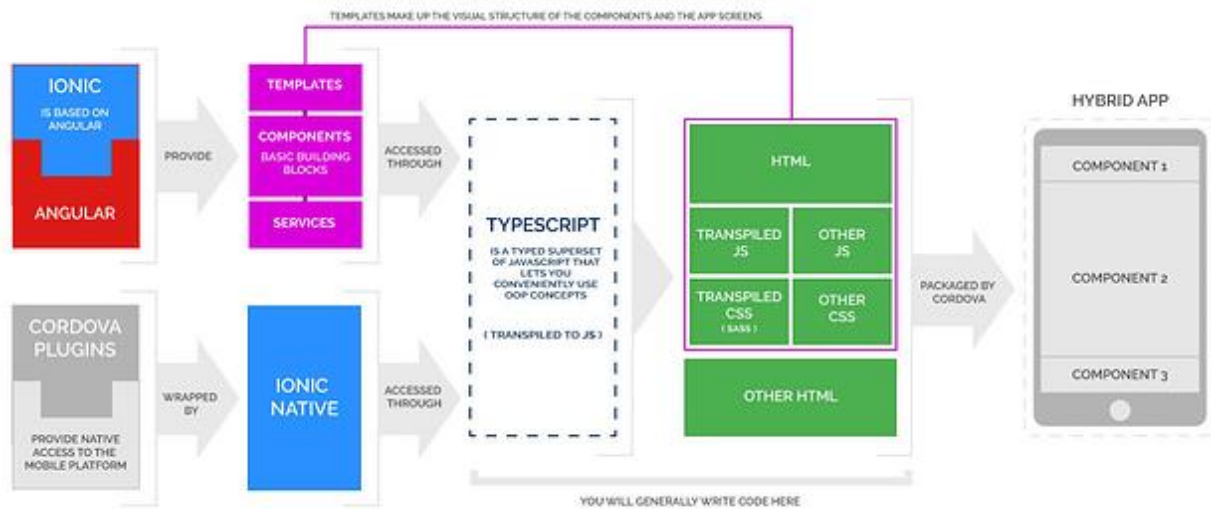


Figura 14: Funcionamiento de Ionic.

Fuente: Foro de Ionic Framework.

11.3. Lenguajes usados con Ionic

11.3.1. HTML

El lenguaje HTML hace referencia a las siglas en inglés “*HiperText Markup Language*”, que significa “lenguaje de marcas de hipertexto”. Se trata de un lenguaje de marcado para la creación de páginas web. Un lenguaje de marcado:

“Es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación”.¹³

Por tanto, no se trata de un lenguaje de programación puesto que no incorpora la declaración de variables ni funciones aritméticas.

Es un estándar a cargo del Consorcio WWW (*World Wide Web Consortium (W3C)*), organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobretodo en lo referente a su escritura e interpretación. Se trata del lenguaje web más importante que ha sido adoptado por todos los sitios web y los navegadores actuales. Intenta ser un lenguaje que permita que cualquier página web pueda ser

¹³ Fuente: Wikipedia.

interpretada de la misma manera por cualquier navegador web, aunque aún difieren las visualizaciones entre un navegador y otro de una misma página cuando no han sido actualizados para poder ver una versión de HTML más reciente.

El diseño en HTML, aparte de cumplir con las especificaciones propias del lenguaje, debe respetar ciertos criterios de accesibilidad web, siguiendo las normativas y leyes vigentes en los países donde se regule dicho concepto. Estos criterios se encuentran disponibles y son desarrollados por el W3C a través de las Pautas de Accesibilidad al Contenido Web, aunque muchos países tienen especificaciones propias, como es el caso de España con la Norma UNE 139803¹⁴.

Este lenguaje define la estructura básica de la página web, en nuestro caso de la aplicación, y a través de un código define su contenido como imágenes, texto, campos de entrada, botones, etc. Este lenguaje se escribe a través de etiquetas introducidas en corchetes angulares (`< > Contenido </>`). En su interior se define el tipo de etiqueta que se va a introducir y entre ambos se encuentra el contenido de dicha etiqueta.

La estructura de una página de HTML de nuestra aplicación consta de una cabecera y un contenido, por lo que, en cierta manera, este lenguaje le da forma a la aplicación. En la cabecera se escriben los títulos y en el contenido, el resto.

En las páginas de HTML se puede incluir tanto código de JavaScript como código CSS por medio de unas etiquetas específicas. Dado que en esta aplicación se cuenta con una página específica para introducir estos tipos de código, no será necesario mezclarlos y se colocará cada código en su página correspondiente. Solo se introducirá color dentro del HTML en los botones ya que cuentan con una propiedad llamada “color” y, por tanto, es mucho más fácil asignarlo en la propia definición del botón (no debería estar definida en HTML sino en el archivo CSS).

Prácticamente todas las etiquetas se abren y se cierran igual, pero con una barra (/). Por ejemplo, en el caso del texto de la cabecera, sería de la siguiente manera: `<h1>` `</h1>`, `<h2>` `</h2>`, etc., La excepción son algunas etiquetas que no necesitan ser cerradas como `
` que sirve para introducir un salto de línea.

En nuestra aplicación se han utilizado las etiquetas siguientes:

¹⁴ Norma UNE 139803: “Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad para contenidos en la Web”.

Etiqueta	Uso
<code><ion-header> </ion-header></code>	En su interior va el contenido de la cabecera de la página.
<code><ion-navbar> </ion-navbar></code>	Incorpora código que se va a ver afectado por la navegación entre páginas.
<code><ion-title> </ion-title></code>	Sirve para colocar el título a la página
<code><ion-content> </ion-content></code>	Muestra el contenido de la aplicación.
<code><ion-list> </ion-list></code>	Crea una lista.
<code><ion-item> </ion-item></code>	Sirve para introducir <i>inputs</i> , ya sean como lista, para introducir texto o de cualquier otro formato.
<code><ion-label> </ion-label></code>	Etiqueta que se le incorpora al <i>input</i> .
<code><ion-option> </ion-option></code>	Define una de las opciones de una lista.
<code><ion-select> </ion-select></code>	Crea una lista dinámica al pulsar sobre el <i>input</i> .
<code><ion-input> </ion-input></code>	Crea un input de texto en el que el usuario escribe.
<code><button> </button></code>	Crea un botón.
<code><ion-icon> </ion-icon></code>	Introduce un icono. Si está dentro de las etiquetas de <i>button</i> , será un botón de icono solo.
<code><ion-buttons end> </ion-buttons></code>	En su interior se define uno o varios botones. Es útil utilizando la propiedad <i>end</i> o <i>start</i> antes del cierre del primer corchete angular, pues los iconos de su interior estarán al principio o al final del título o donde esté colocado, respectivamente.
<code><ion-toggle> </ion-toggle></code>	Se trata de un botón dinámico que al pulsarlo se mueve hacia la derecha o izquierda, activándose o desactivándose respectivamente.

<code><div> </div></code>	Sirve para definir un espacio de la aplicación, en donde irá colocado lo que se defina en su interior. Es para líneas de texto u objetos con grandes dimensiones.
<code> </code>	Es un elemento de línea, es decir, se comporta como un texto lo que se coloque en su interior. Es para líneas de texto pequeñas.
<code><p> </p></code>	Es un elemento de bloque. Se comporta como una caja.
<code><ion-scroll> </ion-scroll></code>	Activa o desactiva, según la propiedad que se le añada, el desplazamiento en el mapa.
<code><h3> </h3></code>	Título con la tercera mayor letra.
<code><ion-card> </ion-card></code>	En su interior se define contenido visual, como por ejemplo imágenes.
<code><hr> </hr></code>	Representa un cambio de párrafo.
<code></code>	Introduce una imagen
<code>
</code>	Salto de línea.

Tabla 1: etiquetas de HTML usadas en la aplicación

11.3.2. JavaScript (TypeScript)

JavaScript es un lenguaje de programación interpretado, es decir, que no necesita compilación, sino que es interpretado por un navegador web. Es utilizado principalmente para ejecutar acciones del lado del cliente (usuario). Su principal objetivo es agregar interactividad a las páginas web, permitiendo incorporar efectos dinámicos a las páginas para una mayor vistosidad. Está definido por un lenguaje de programación orientado a objetos (POO) y se basa en prototipos; es decir, no existen las clases, y por ello el concepto de herencia se lleva a cabo de otra manera. Además, al igual que en otros lenguajes de programación, se trata de un lenguaje débilmente tipado, puesto que no necesita indicar el tipo de datos de una variable. No hay que

confundirlo con Java ya que, aunque adopte nombres y convenciones de programación de este, tienen semánticas y propósitos diferentes.

Dentro del JavaScript se encuentra el TypeScript, que es el lenguaje que se utiliza en Ionic a partir de la segunda versión de este. Este tipo de lenguaje es libre y de código abierto, ya que se accede al código de manera libre pudiendo modificarlo. Es desarrollado y mantenido por Microsoft y los archivos que se generan son con la extensión “.ts”. Las diferencias con JavaScript es que incluye el tipado estático (definir el tipo de las variables y de los valores que devuelven las funciones) y los objetos basados en clases (mejora la POO). Ambos utilizan la misma sintaxis.

TypeScript es por tanto un superconjunto de JavaScript. Esto quiere decir que incluye tanto JavaScript como EcmaScript 6, un lenguaje posterior a este con mejoras, como por ejemplo mejor orientación a objetos. TypeScript engloba estos dos lenguajes y además añade otros elementos enriqueciendo el lenguaje.

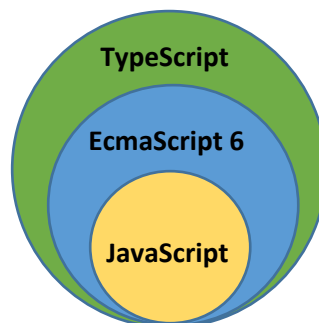


Figura 15: lenguajes que engloba el TypeScript.

TypeScript y JavaScript son totalmente compatibles ya que todos los archivos “.ts” se convierten a código “.js” (JavaScript).

Al tener la característica de tipado estático, el código se hace mucho más legible para terceros comparado con JavaScript, puesto que se conoce el tipo de variable que se está utilizando y el que devuelven las funciones. Además, otra de sus ventajas es que puede utilizar todas las características avanzadas de la programación orientada a objetos y es compatible con cualquier navegador.

TypeScript resuelve todos los problemas de JavaScript relacionados con aplicaciones grandes ya que con TypeScript se detectan los errores en tiempo de escritura, por lo que aumenta la productividad, y crea un código más legible y más ordenado, obteniendo así una mejor mantenibilidad, debido por ejemplo al uso de clases.

Este tipo de lenguaje es el que indica el contenido, las funciones, declara las variables, etc. Utiliza gran parte de la estructura de programación en C, por ejemplo: *if*, *for* *switch*, ... Para declarar una variable local se utiliza la palabra reservada *var* y cuando es una variable local será *let*.

Como la mayoría de lenguajes, tiene un objeto *console* que sirve para imprimir por pantalla en la consola de depuración el mensaje que se le indique. Se hará por medio de la sentencia *console.log* ("*Texto*", *variable*).

En TypeScript se importan las librerías necesarias en cada página y también se importan otras páginas de la aplicación. Al crear las páginas se importan automáticamente dos, que se muestran a continuación. El resto se van añadiendo a medida que se programa el código.

A continuación, se han copiado los trozos de código referentes a la importación de librerías en TypeScript de nuestra aplicación:

```
import { Component } from '@angular/core';  
import { IonicPage, NavController, NavParams } from 'ionic-angular';
```

```
import { AlertController } from 'ionic-angular';  
import { MapaPage } from '../mapa/mapa';  
import { MenuPage } from '../menu/menu';  
import { CalculoEficienciaPage } from '../calculo-eficiencia/calculo-  
eficiencia';  
import { CalculoCortoPage } from '../calculo-corto/calculo-corto';  
import { Geolocation, Geoposition } from '@ionic-native/geolocation';  
import { InforVehiculoPage } from '../infor-vehiculo/infor vehiculo';  
import { TabsPage } from '../tabs/tabs';  
import { RegistroPage } from '../registro/registro';  
import { HomePage } from '../home/home';  
import { InicioSesionPage } from '../inicio-sesion/inicio-sesion';
```


Como hemos dicho, TypeScript es un lenguaje de POO y tiene clases. Por tanto, la estructura de las páginas de este lenguaje consta de la declaración de una clase la cual cuenta con todo el contenido de TypeScript. Tiene un constructor y tras él se definen el resto de métodos de la página.

El constructor es un método que se ejecuta nada más cargar la página y su mayor uso es inicializar las variables. Por tanto, estas se declaran en la clase por fuera del constructor y se inicializan en su interior. El valor que se les da desde dentro del constructor se cambiará o no desde el resto de métodos definidos en la clase. A estas variables que se declaran en la clase, fuera de cualquier método, se accede por medio de la palabra reservada *this* que apunta directamente a la variable deseada de la propia clase.

Este lenguaje no importa las variables de una página a otra, sino que hay que hacerlo manualmente y esto es posible gracias al servicio *navParams* de Ionic, a parte del servicio *navController* que permite la navegación entre páginas. Ambas se definen como argumentos públicos en el constructor. Para poder importar dichas variables, se define un método que se ejecutará al hacer clic sobre un botón. Este método estará compuesto en su interior por el acceso al servicio *navCtrl* con su propio método *push* para enviar dichos parámetros a la página de destino que se coloque después de este método. El proceso es el siguiente: se definen en la clase fuera del constructor las variables que se desean exportar a otra página, se declara el nuevo nombre que le daremos a la página de destino y se inicializa dentro del constructor esta variable que hemos declarado con el nombre nuevo igualada al nombre real de la página de destino. Posteriormente, en el interior del método que se ejecuta al hacer clic sobre el botón, accedemos al servicio *this.navCtrl.push(this.NombreNuevoDeLaPagina, {...})*. En el interior de las llaves irá el nombre que le daremos a cada una de las variables que vamos a exportar, seguido de dos puntos y el acceso a la variable definida al inicio en la clase a través de la palabra reservada *this*. Todas las variables a las que se le asigne un nuevo nombre aquí, irán seguidas por una coma excepto la última. En la página de destino, se declararán las variables que se utilizarán, que serán los nombres nuevos que le hemos asignado en la página de inicio. Por último, dentro del constructor, se accede a dichas variables con *this* y se inicializan con *navParams.data.variable*. De esta manera, estas variables que se han asignado en el constructor, son las mismas que se habían definido

en la página de inicio, por lo que se importarán sus valores para poder trabajar con ellos en esta página de destino.

Para ilustrar el uso de este hecho en nuestra aplicación, se ha copiado y se muestra a continuación, los trozos de código pertenecientes a la definición del botón y el método que se ejecuta al darle a este y todo el proceso que le sigue para poder importar las variables de una página a otra:

```
<button id="boton_okC" ion-button round color="secondary"
(click)="Iralmapa()">ACEPTAR</button>
<!--Al hacer clic sobre el botón, se ejecutará el método Iralmapa()-->
```

```
//Declaración en el constructor como argumentos de entradas navCtrl y
navParams.
constructor(public navCtrl: NavController, public navParams:
NavParams){...}
```

```
//Declaración de variables, que van a ser importadas, dentro de la
clase pero fuera del constructor.
```

```
mapaPage : any;//Nuevo nombre que se le dará a la clase de destino.
anchoC : any;
altoC : any;
personasC : any;
pesoC : any;
capacidadC : any;
intervaloC : any;
cdC : any;
nom : any;
```

```
Iralmapa(){ //Definición del método.

this.navCtrl.push(this.mapaPage,{
anchoC1: this.anchoC,
altoC1: this.altoC,
capacidadC1 : this.capacidadC,
personasC1 : this.personasC,
pesoC1 : this.pesoC,
intervaloC1 : this.intervaloC,
cdC1 : this.cdC,
nombre : this.nom
});
```

```
}
```

```
//Declaración de las variables en la página de destino en la clase,  
pero fuera del constructor.
```

```
anchoC1 : any;  
altoC1 : any;  
personasC1 : any;  
pesoC1 : any;  
capacidadC1 : any;  
intervaloC1 : any;  
cdC1 : any;  
nombre : any;
```

```
//Dentro del constructor de la página de destino, inicialización de las  
variables importadas con navParams.
```

```
this.anchoC1=navParams.data.anchoC1;  
this.altoC1=navParams.data.altoC1;  
this.personasC1=navParams.data.personasC1;  
this.pesoC1=navParams.data.pesoC1;  
this.capacidadC1=navParams.data.capacidadC1;  
this.intervaloC1=navParams.data.intervaloC1;  
this.cdC1=navParams.data.cdC1;  
this.nombre=navParams.data.nombre;
```

Una de las nuevas incorporaciones a TypeScript es la sustitución de la palabra *function* por “=>”. Este cambio produjo errores en el desarrollo del código debido al desconocimiento de su existencia. Al introducir la palabra *function* para declarar una función, la palabra reservada *this* que apunta a las variables declaradas en la clase normalmente, en este caso dejan de apuntar a dicha clase y apuntarían a otra distinta que crea, lo que puede ser una gran ventaja en ocasiones, pero en nuestro caso supuso errores pues al intentar acceder a estas variables que se usaban como *this.latit* y *this.long* desde otra función, aparecían indefinidas pues no estaban siendo modificadas las variables de la propia clase sino de otra clase. En definitiva, las funciones con flecha, así se les llama al uso de “=>”, permiten el uso del *this* en el contexto actual. Su uso es reemplazar la palabra *function* por “=>” justo después de los paréntesis con los argumentos de entrada si los hubiera.

```
//Utilización de la palabra function en versiones anteriores.
```

```
setTimeout(function() {  
  
});
```

```
//Actualización y forma de uso en el código de la aplicación.  
  
setTimeout(() => {  
  
});
```

11.3.3. CSS

El lenguaje CSS hace referencia a las siglas en inglés “*Cascading Stylesheets*”, que significa “Hoja de estilos en cascada”. Se puede definir como:

“Un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado”¹⁵.

Se trata, por tanto, de un lenguaje que da forma, estilo y diseño visual al código creado en HTML de la página web que después se convertirá en nuestra aplicación. Junto con HTML y TypeScript, son tres lenguajes usados para hacer visualmente atractivas las páginas web.

Su aparición fue un evento muy importante ya que supuso la separación entre el contenido del documento y la forma de presentación de este, las capas, los colores, las fuentes, los márgenes, los bordes, ... Esta separación permite el uso del mismo archivo CSS para varios archivos HTML, lo que optimiza el tiempo de trabajo, el tiempo de carga del sitio y mejora la mantenibilidad.

Estos estilos pueden aplicarse directamente al archivo HTML, pero, como ya hemos dicho, en nuestro caso no usaremos esta herramienta pues haremos uso del propio archivo CSS para definirlos. Esta técnica es la más recomendada.

La sintaxis del código consta de reglas. Cada regla está compuesta por el selector del elemento y el bloque de código entre llaves, llamado bloque declarativo. Este selector es el elemento que indica a qué parte del código de HTML se aplica la regla y

¹⁵ Fuente: Wikipedia.

el bloque declarativo va entre llaves y está compuesto por declaraciones, las cuales consisten en una propiedad a la que se le asigna un valor.

Unas de las declaraciones más importantes en la aplicación en este lenguaje son la del ancho y el alto del mapa, ya que sin ellas no se visualiza el mapa. A continuación, se muestra el código utilizado para ello:

```
#map {
  width: 100%;
  height: 100%;
  .
  .
  .
}
```

Las propiedades usadas en este lenguaje en la aplicación son:

Propiedad	Uso
Width	Ancho
Height	Alto
Opacity	Opacidad
Transition	Tiempo entre ejecuciones de tareas
Margin	Margen con el borde de la aplicación
Background-color	Color de fondo
Border-radius	Radio del borde
Border	Tamaño del borde
Box-shadow	Color de sombra de la caja
Box-sizing	Tamaño de la caja
Font-family	Tipo de fuente
Font-size	Tamaño de la Fuente
Font-weight	Ancho de la fuente

Tabla 2: propiedades de CSS usadas en la aplicación.

Mediante estos tres lenguajes se creará la aplicación. Como conclusión a las definiciones se puede decir que el lenguaje HTML sirve para indicar qué estructura tendrá la aplicación, TypeScript le da contenido y define las operaciones que se llevarán a cabo y CSS le da forma y estilo a la estructura definida en HTML.

11.4. APIs

Como su propio nombre indica, las APIs son interfaces de programación de aplicaciones. Son el medio por el cuál un software se comunica o interactúa con otro y permiten la creación de programas específicos sin partir desde cero, sino que son la base sobre la cual se establecen dichos programas. Además, estas interfaces no se muestran al usuario en su totalidad, sino que este solo ve una cara de ellas, el resto se hace internamente. Por ejemplo, si en una aplicación se pulsa un botón para compartir una información en dicha plataforma que indica el botón, el usuario no ve la comunicación entre la aplicación sobre la que está navegando y en la que quiere compartir la información, sin embargo, se están comunicando y compartiendo comandos, funciones y protocolos gracias a las APIs para que se pueda completar la acción ordenada por el usuario sin tener que salir de la propia aplicación.

Se trata de un método de abstracción entre los componentes del software ya que se conoce lo que hace la API, pero no de la manera que lo hace. Su propósito es proporcionar una serie de funciones de uso general, permitiendo así que el usuario no programe desde el principio, sino que este sea su inicio. De esta manera, se incorpora código que ya está probado y ya se sabe que funciona correctamente.

No solo existen APIs de aplicaciones de redes sociales u otras plataformas, como las del ejemplo, que son prácticamente nuevas, sino que existen otras para comunicarse con el sistema operativo, con bases de datos o con protocolos de comunicaciones. Por ejemplo, desde un ordenador, con un sistema operativo dado, que quiera conectarse a Internet, se comunicará con la API del sistema operativo para reconocer la tarjeta de red: no tiene que incluir el usuario funciones en el código para dicho reconocimiento.

En la aplicación objeto de este proyecto, se hace uso de la API de Google Maps. Mediante ella se puede visualizar el mapa, incorporar el cuadro de búsqueda y sugerencias de lugares cercanos, los marcadores del sitio actual y el de destino con la propiedad de que puedan arrastrarse por el mapa, las indicaciones de cómo llegar al punto de destino junto con la distancia total del trayecto y el tiempo aproximado que se tardará en recorrerlo, y trazar la ruta entre dos puntos. Estas son sólo algunas de las funciones que hemos utilizado de esta potente herramienta, ya que existen muchas más.

Se hará en TypeScript incorporando la librería de Geolocalización y también en el constructor como argumento, tal y como se muestra en el siguiente trozo de texto:

```
import { Geolocation, Geoposition } from '@ionic-native/geolocation';

constructor(public navCtrl: NavController, public navParams: NavParams,
public geolocation: Geolocation)
```

Para poder hacer uso de esta API, es necesario activar el GPS en el servidor en el que se ejecuta el programa, para que así pueda obtener la ubicación actual del usuario. En el código, para saber si está activado y ha obtenido bien la localización, se hará por medio de la propiedad de Geolocalización llamada *getCurrentPosition()*.

Posteriormente, se creará el mapa y el marcador de posición mediante unas nuevas variables pertenecientes a Google Maps. Para ello habrá que declarar la variable Google a la que se accede en las declaraciones anteriormente mencionadas. Se declara fuera de la clase, justo después de los *imports* de otras librerías. Quedarían de la siguiente manera:

```
import { Geolocation, Geoposition } from '@ionic-native/geolocation';

declare var google;
```

```
this.map = new google.maps.Map
let marker = new google.maps.Marker
```

Se declararán de la misma manera las variables que permite las sugerencias de lugares cercanos y el cuadro de información del lugar de destino:

```
var autocomplete = new google.maps.places.Autocomplete(input);
var infowindow = new google.maps.InfoWindow();
```

La función en la que están definidas estas variables tiene como parámetro de entrada la posición actual dada por el GPS del dispositivo:

```
loadMap(position: Geoposition){...}
```

Gracias a este argumento, se podrá acceder a la latitud y longitud de la ubicación para poder pintar el marcador sobre ella y poder trabajar con estas coordenadas posteriormente, y se hace de la siguiente forma:

```
let latitude = position.coords.latitude;  
let longitude = position.coords.longitude;
```

Así es como se ha incorporado el sistema de Geolocalización de la API de Google Maps y se han declarado las variables que se utilizan para poder incorporar un mapa en la aplicación con un marcador de la posición actual.

Para poder trazar la ruta entre dos puntos, se utilizarán las siguientes estancias de Google Maps:

- **DirectionsService:** Se trata de un servicio al que se le envían unos parámetros para calcular la ruta sugerida por Google desde el punto A hasta el punto B y devuelve una serie de datos que trata **DirectionsRenderer**.
- **DirectionsRenderer:** Este servicio se encarga de mostrar en nuestro mapa la ruta con sus marcadores y un panel con las indicaciones que tendrá que seguir para llegar al punto de destino.

```
this.directionsService = new google.maps.DirectionsService();  
this.directionsDisplay = new google.maps.DirectionsRenderer();
```

En la función de trazar la ruta, existen varias opciones que se pueden incorporar para mandar datos a Google y que los interprete. Los que se han utilizado son: un punto de origen y otro de destino, el modo en el que se va a recorrer la ruta que en nuestro caso es siempre en coche (“DRIVING”) y elegida la opción de evitar peajes. Existen más opciones como los *waypoints*, que son puntos intermedios por los que debe pasar la ruta, pero en este caso no se han utilizado.

```
this.directionsService.route({  
  origin: ...,  
  destination:...,  
  travelMode:...,  
  avoidTolls: true  
}, (...,...) {
```




Esta función siempre tiene que tener una opción para saber si la respuesta que ha dado es buena o no. Si es buena, se mostrará la ruta y las indicaciones que hay que seguir. Si no, se mostrará un mensaje de error por pantalla. Esto se indica en el interior de las segundas llaves del código mostrado anteriormente.

11.5. Modos de funcionamiento

Para acceder a cualquiera de los dos modos disponibles, se hará desde la página de inicio (**apartado 11.6.**), donde existe un botón para cada modo de funcionamiento (ver figura 18). Al pulsar sobre alguno de ellos, llevará al usuario a una nueva página, dependiendo del modo que se haya escogido.

Se fijarán dos modos de comenzar una vez se accede a la aplicación:

- Modo de calibración inicial:

Este modo sirve para comparar los datos obtenidos con la propia aplicación con los proporcionados por la aplicación del vehículo, ya que algunos de los parámetros que se utilizan en la aplicación se obtienen de la base de datos, pero otros no, por lo que hay que obtenerlos experimentalmente. Es por ello que existe este modo, que permite la obtención experimental de algunos datos para la posterior utilización en el funcionamiento normal.

En este tipo de funcionamiento, se le pedirán todos los parámetros necesarios al usuario. Los datos resultantes de la calibración permanecerán almacenados en memoria mientras no se vuelva a activar este modo de funcionamiento.

Para ejecutar el modo de calibración, será necesario fijar un intervalo de distancia constante de 1 kilómetro. Este intervalo de distancia constante será el número por el que se dividirá la distancia total a recorrer, obteniendo una serie de intervalos en los que se

calculará la altura con respecto al mar. Se escoge 1 kilómetro ya que es una distancia favorable para los cálculos, ya que una distancia más pequeña supondría pérdida de tiempo y con una mayor se perdería información. Para conocer el tiempo, se introducirá un reloj interno en la aplicación que comenzará a contar una vez el usuario le da a comenzar la calibración después de trazar la ruta sobre el mapa: en este momento se supone que el vehículo comienza a moverse. Sabiendo la distancia que se va recorriendo y el tiempo que tarda, sabremos la velocidad. La distancia se obtendrá por medio de una API de Google Maps que nos proporciona todas las indicaciones que hay que seguir para llegar al destino y también la distancia a recorrer.

En el interior de esta aplicación, el modo de calibración está programado de manera que, al presionar el botón de “Calibrar” desde la página de “Mapa”, se comenzarán a calcular todas las pérdidas y ganancias que tiene el vehículo a medida que avanza de la siguiente manera: la aplicación recibe como *inputs*, por parte del usuario, el modelo del coche y con él, el ancho, el alto, el C_x , el peso del vehículo y el número de pasajeros. Con el ancho y el alto se calcula la superficie frontal, y con el peso del vehículo y el número de pasajeros, suponiendo un peso medio de cada persona unos 80 kilogramos, se obtiene el peso total del vehículo.

Antes de calcular ningún parámetro, se le pasará por medio del GPS del dispositivo utilizado la posición actual del usuario, que se mostrará en el mapa. La posición que se elija como destino se guardará en variables para poder calcular su longitud y latitud, y así poder dibujar en el mapa el marcador correspondiente. A través de la API de Google Maps se obtiene la distancia total del trayecto y se guarda en una variable. A través de este sistema, se obtiene el perfil de alturas a partir del cálculo de altura cada intervalo constante de 1 km. Todos los valores de *inputs*, el peso y la superficie calculadas, la distancia total y la altura, serán necesarios para el cálculo de las energías que se calcula a continuación. También será necesaria la velocidad, que en este modo de funcionamiento se obtiene dividiendo el intervalo constante de distancia (1 km), entre el tiempo que esté guardado en el cronómetro justo hasta el momento en el que el vehículo haya recorrido el kilómetro.

Se hacen internamente todos los cálculos descritos en el **apartado 10** del estudio energético y, una vez se termina el proceso de calibrar, se mostrará la cantidad de kWh que se han consumido a lo largo del recorrido y la energía que habrá que cargar en la

batería para poder realizar el trayecto. Este es el objetivo de este modo de funcionamiento: compara los valores obtenidos y extraer del propio software proporcionado por el vehículo, valores que nos servirán posteriormente. El coeficiente de resistencia a la rodadura utilizado en este modo de funcionamiento es una aproximación que se hace según las condiciones de la carretera que se vaya a tomar [7].

Desde que concluya el tramo de prueba, se comprueban los resultados obtenidos con los correspondientes a la aplicación propia del coche. Desde aquí debemos extraer algunos de los valores que nos interesan para nuestra estimación posterior, como por ejemplo el coeficiente de resistencia a la rodadura. Se guardarán los datos obtenidos en el usuario que se haya registrado hasta que se realice una nueva calibración. En principio la calibración sólo será necesaria una vez mientras no cambien las condiciones de funcionamiento del coche.

- Modo de funcionamiento normal:

Una vez terminada la calibración y comparado los resultados obtenidos, el usuario querrá comenzar un trayecto. Es por ello que desde que el usuario haga clic sobre el botón de “terminar calibración”, terminará ese proceso y le redirigirá a una nueva página perteneciente ya al modo de funcionamiento normal.

Posteriormente la aplicación irá hacia la página del mapa donde, al igual que sucedía en el modo calibración, el usuario podrá seleccionar sobre el mapa el recorrido a realizar.

Uno de los valores a utilizar para predecir el alcance del coche es la velocidad en cada tramo. Naturalmente, como todo proceso “predictivo” tenemos que “estimar” dichos valores, puesto que no son conocidos a priori. En nuestro caso, la velocidad será supuesta desde el principio y constante. Su valor será elegido por el usuario según las carreteras por las que vaya a circular. Esta es la opción elegida para la primera versión de esta aplicación. Existen otras opciones, como obtener datos de las carreteras por las que va a circular y adaptar la velocidad supuesta constante a la velocidad máxima de cada tramo, cuyo efecto sobre la fiabilidad de la aplicación habría que analizar para futuras versiones (**apartado 12.**).

La diferencia de este modo de funcionamiento con el anterior, es que el primer output que obtendrá el usuario será la energía que consumirá el vehículo durante el trayecto marcado, antes de que el coche haya empezado a circular, a diferencia del modo de “calibración”, que el usuario obtenía las pérdidas ocasionadas durante el trayecto una vez finalizado este. Además, también sabrá la energía que tendrá que cargar en la batería para poder consumir dicha energía perdida. Además, este modo de funcionamiento cuenta con un C_{rr} actualizado, pues se utiliza el proporcionado por el propio vehículo, ocasionando un resultado más óptimo.

Este modo de funcionamiento está programado de la siguiente manera: la aplicación obtiene como *inputs* internamente todos los valores que se le pidieron al usuario en el modo de “calibración” y, además, se obtendrá también como input el C_{rr} , el C_x y el número de pasajeros actualizado. Con todos estos datos, el modo de proceder interiormente en cuanto a código es igual que en el de “calibración”. La única diferencia se encuentra en el cálculo del balance energético que, para poder hacer la predicción, no se hace uso del crono, sino que se obtiene el tiempo a través de la relación entre la velocidad y la distancia. Se obtendrá como *output* la predicción de energía consumida que se tendrá en el trayecto indicado y se mostrará en la pantalla de la aplicación. Otro output será, de igual manera, la energía que sea necesaria recargar para poder realizar el trayecto.

11.6. Funcionamiento de la aplicación

Como ya hemos dicho, Ionic se compila a través de un servidor. Esto se hace a través del terminal que se esté usando, introduciendo: *ionic serve*. Ya que se trata de un *framework* multiplataforma, también ofrece la posibilidad de ver la apariencia de la aplicación en Android, iOS y Windows. Para ello se introduce en el terminal: *ionic serve --lab*. Cada uno de los softwares tiene sus propias características, por lo que, al introducir por ejemplo un botón en una página, en una plataforma se verá a un lado y en otra a otro lado.

En esta página se pueden distinguir dos zonas diferentes de acceso a la aplicación que describimos a continuación: el menú a pie de página y el menú principal.

11.6.1. Menú a pie de página

En la portada de la aplicación encontramos un menú horizontal (ver figura 16) que nos permite acceder a cuatro páginas diferentes.



Figura 16: apariencia del menú horizontal inferior de la aplicación

- **Página “Inicio”**

En la página de “Inicio” se incluye el logo de la aplicación y tres botones para poder acceder a uno u otro modo de funcionamiento.



Figura 17: apariencia de la página de “Inicio” en Android, iOS y Windows respectivamente (parte I).

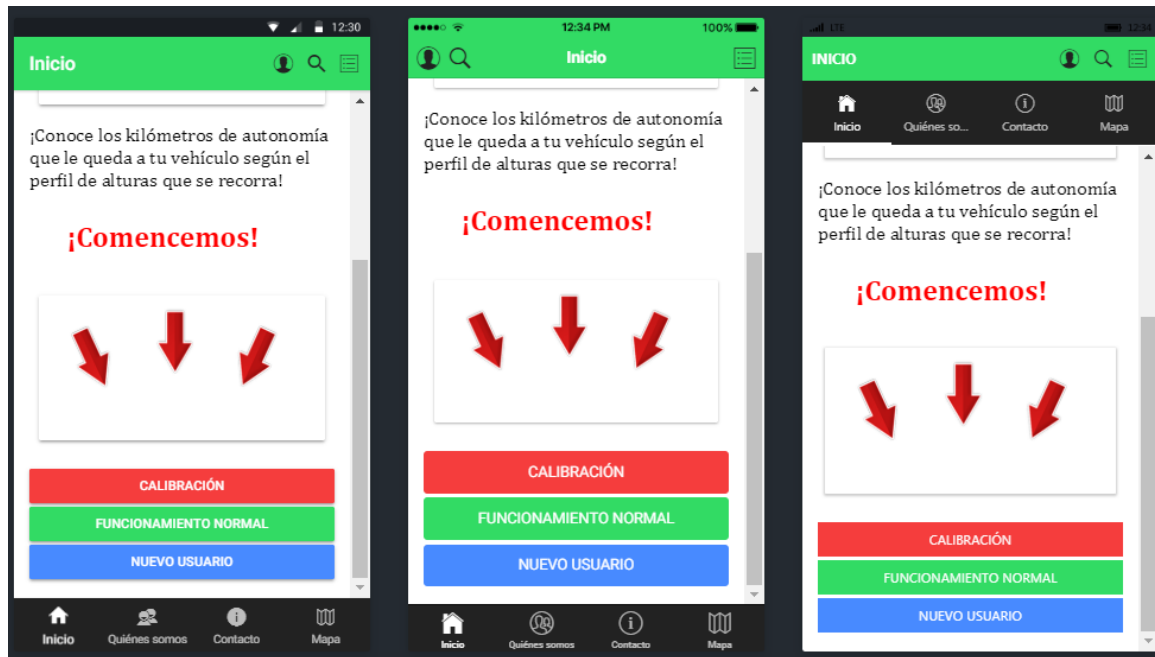


Figura 18: apariencia de la página de “Inicio” en Android, iOS y Windows respectivamente (parte2).

En la parte inferior de la página inicial, se encuentran tres botones: “Calibración” y “Funcionamiento normal” permiten el acceso a los modos de funcionamiento de los que dispone la aplicación, y “Nuevo usuario” sirve para registrarse como tal si es la primera vez que se usa la aplicación.

- Calibración

El proceso que sigue esta página es el siguiente: al acceder a este modo desde la página de inicio, se abrirá una nueva llamada “Calibrar” (ver figuras 19, 20, 21). En ella se elige el modelo de vehículo si está en la lista y si no, se elegirá la opción de “Otro”, añadiendo manualmente el usuario los parámetros que se le piden, obteniéndolos de la ficha técnica de su vehículo. Si elige cualquier otro vehículo de la lista, se le colocarán automáticamente dichos valores, aunque el usuario podrá cambiarlos si cree que no son correctos.

Al finalizar, se hace clic en el botón de aceptar y aparecerá la página con el mapa (ver figura 22) y en él el marcador que indica la posición actual del usuario. Por defecto sale marcada la opción de evitar peajes, pero el usuario podrá desactivarla si así lo desea. En el mapa hay un cuadro para buscar el lugar de destino, ya sea un sitio físico, una dirección o una ciudad (ver figura 23). El usuario dará clic sobre la sugerencia que se adapte a sus necesidades de búsqueda y en ese momento

saldrá en el mapa un marcador con la posición de destino, abriéndose en él un cuadro de información, en el que se indica un número identificador que es propio de cada lugar y único, el código postal del lugar, localidad, provincia y país al que pertenece. Si al buscar el lugar de destino, el marcador se sitúa en una localización que no se cree que es la correcta por el usuario, este podrá arrastrarlo por el mapa hasta la posición que considere adecuada. Esta posición final será la que la aplicación tomará como destino. También podrá hacer lo mismo con el marcador de la posición actual si quiere salir desde otro punto que no sea el que este indica.

El siguiente paso será clicar sobre el botón de trazar ruta y entonces aparecerá la ruta propiamente dicha entre el punto de inicio y el de destino (ver figura 24). Justo debajo de los botones aparecerán las indicaciones que tendrá que seguir el usuario para llegar al punto de destino y la distancia total a recorrer (ver figura 25).

Al pulsar sobre el botón de calibrar, se abrirá una página en la que se mostrará la superficie frontal obtenida a partir del ancho y el alto proporcionados, y el peso total sumando el peso del vehículo más el peso aproximado referido al número de pasajeros que viajen. Además, aparecerá la cuenta del cronómetro de la aplicación que comenzará a contar desde que se entre en la página en la que se encuentra. Una vez se termine el tramo de prueba, se le dará al botón de “terminar calibración” y acabará este modo de funcionamiento (ver figuras 26 y 27).

El criterio que se sigue en este y en el otro modo de funcionamiento disponible, es el cálculo de todas las energías que influyen en el movimiento de un vehículo: energía cinética, potencial, aerodinámica y de rodadura (**apartado 10.5.**)

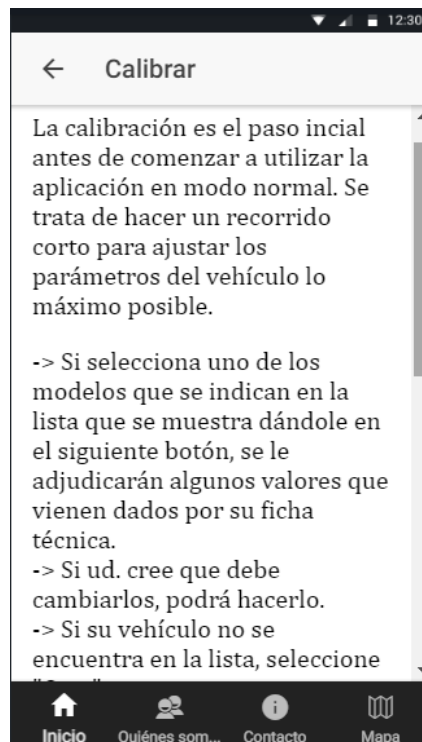


Figura 19: página “Calibrar” (parte1).



Figura 20: página “Calibrar” (parte2).

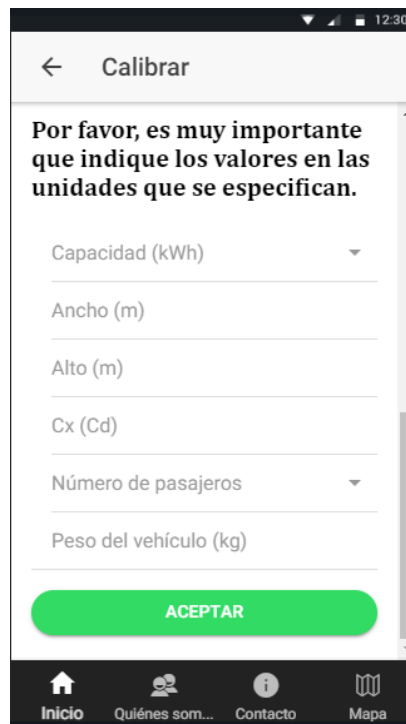


Figura 21: página “Calibrar” (parte3).

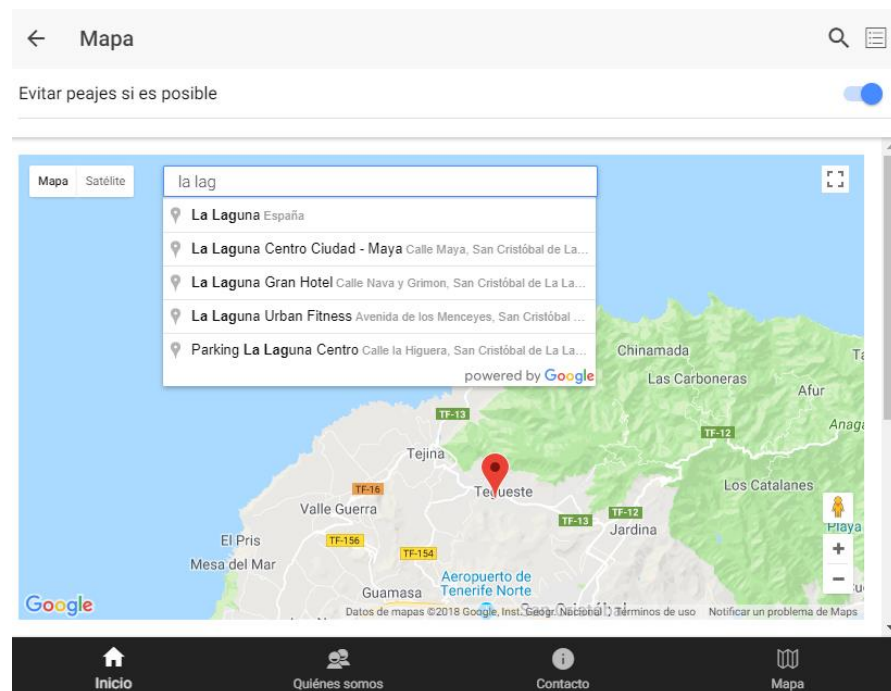


Figura 22: vista de la página del mapa con el marcador inicial y las sugerencias de lugares cercanos del cuadro de búsqueda.

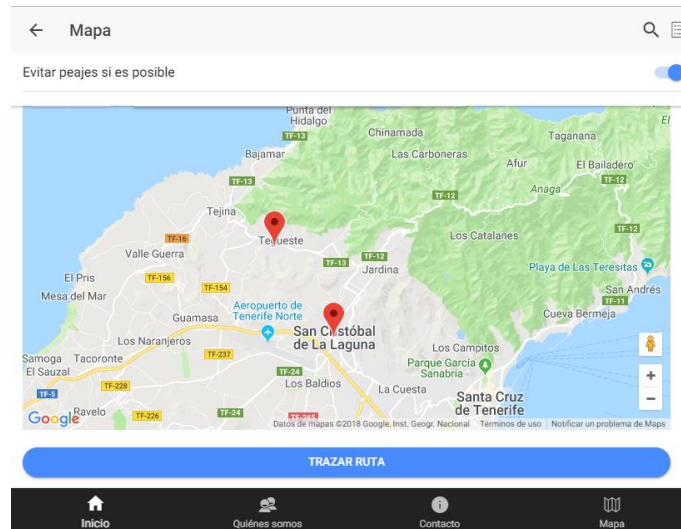


Figura 23: apariencia del botón de “Trazar ruta”.

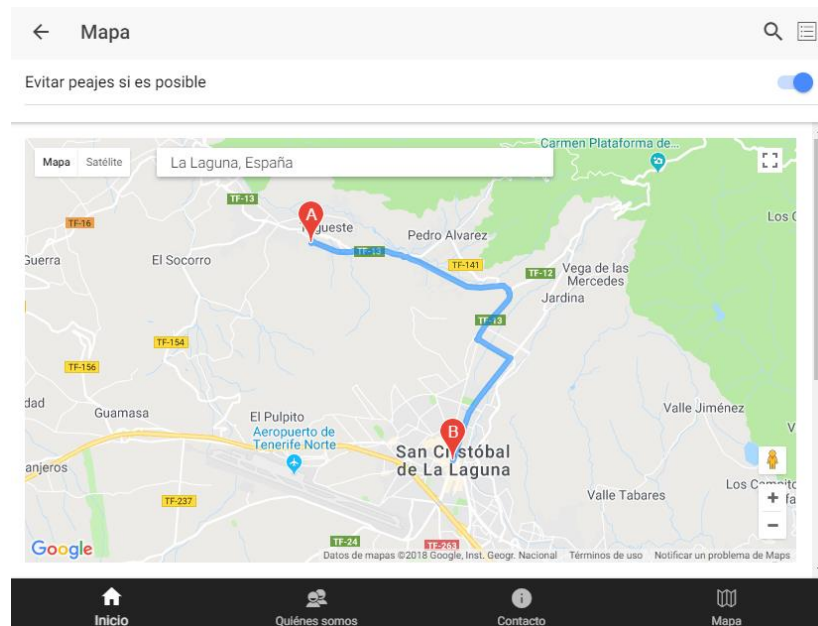


Figura 24: Ruta trazada entre el punto actual y el de destino.

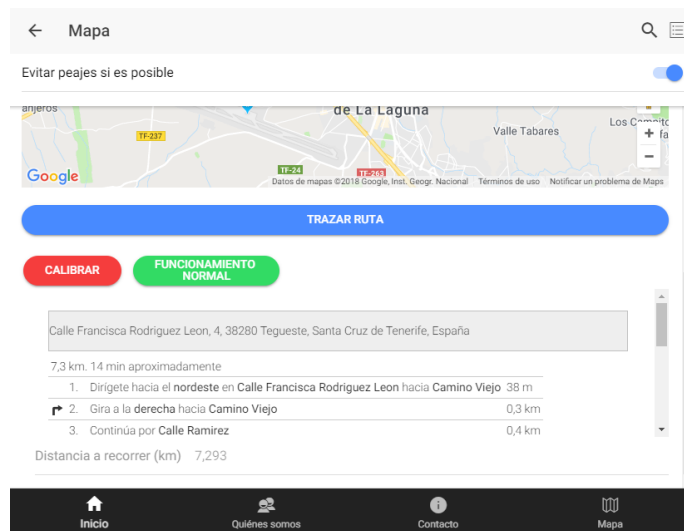


Figura 25: indicaciones para llegar al destino propuesto y botón de “Calibrar”.

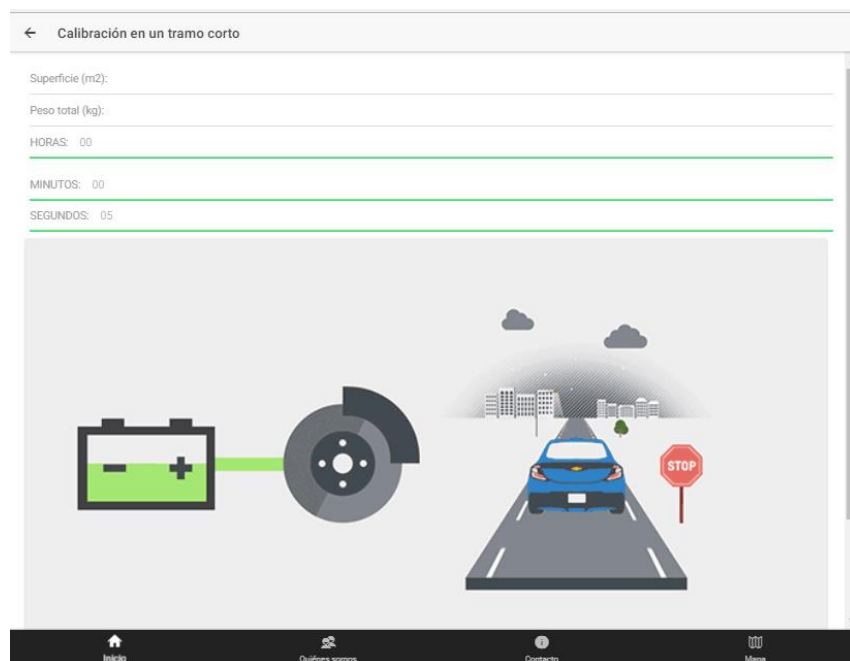


Figura 26: cronómetro durante el proceso de calibración.

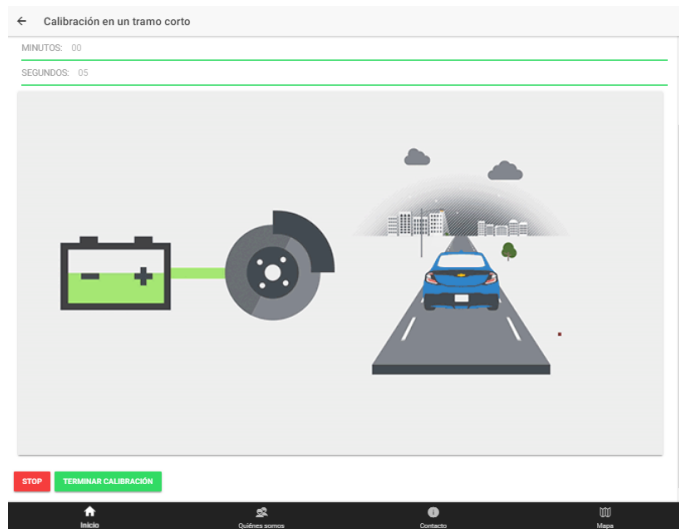


Figura 27: botones para parar el cronómetro y terminar la calibración.

- Funcionamiento normal

El modo de proceder es el mismo que en la calibración. Por lo general, el trayecto en este funcionamiento será más largo y uno de los valores obtenidos tras la calibración se introducirá como *input* en este modo de funcionamiento. Este será el C_{rr} (coeficiente de resistencia a la rodadura: **apartado 9.2.**). El resto de parámetros se importarán internamente partiendo de los introducidos para la calibración previamente, por lo que no será necesario que el usuario vuelva a introducirlos. Por tanto, los inputs de esta página serán el C_{rr} obtenido de la calibración, el C_x (coeficiente aerodinámico: **apartado 9.1.**) por si el usuario desea cambiarlo y el número de pasajeros, ya que puede variar de un modo a otro, a parte de los importados del modo de calibración.

En este modo de funcionamiento normal, siempre aparecerá el modelo de vehículo que el usuario eligió en el modo de calibración, puesto que, si se trata de otro coche, primero ha de completarse una nueva calibración. Si no es la primera vez que accede a la aplicación y ya ha realizado la calibración con el vehículo que pretende utilizar en funcionamiento normal, se indicará el modelo del que se trata en el botón.

Al darle a “Funcionamiento normal”, se abrirá una página que hará lo mismo que en calibración: tendrá un cronómetro que se irá mostrando a medida que avanza y unos botones, uno para parar el cronómetro y el otro para terminar el

funcionamiento normal. Cuando se presiona sobre este último botón, se redirigirá al usuario a la página de “Inicio”.

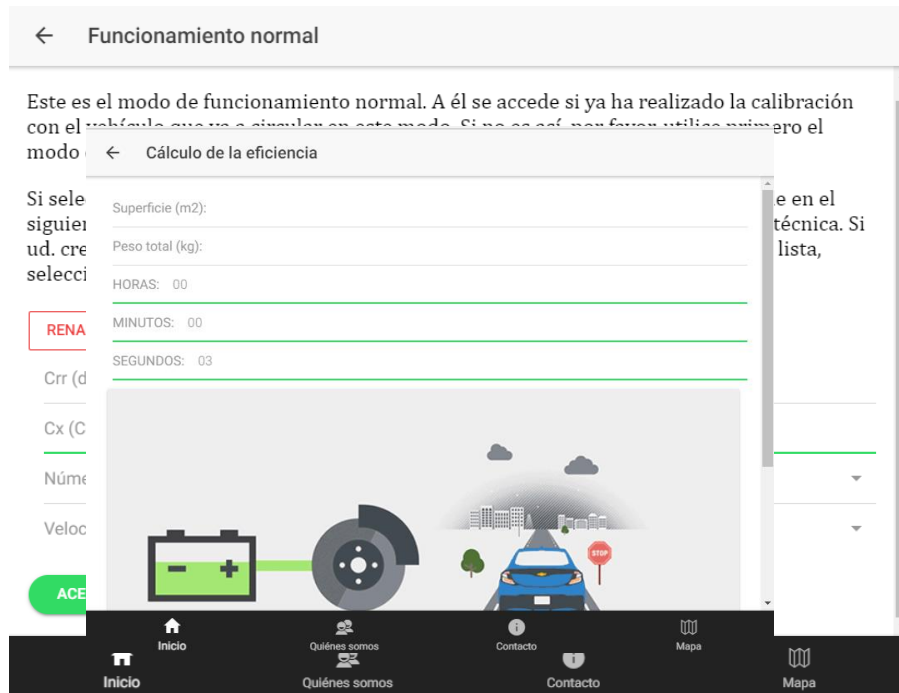


Figura 28: apariencia de la página que aparece al darle a “Funcionamiento normal”.

Figura 29: apariencia de la página donde se calcula la eficiencia en el funcionamiento normal (parte 1).

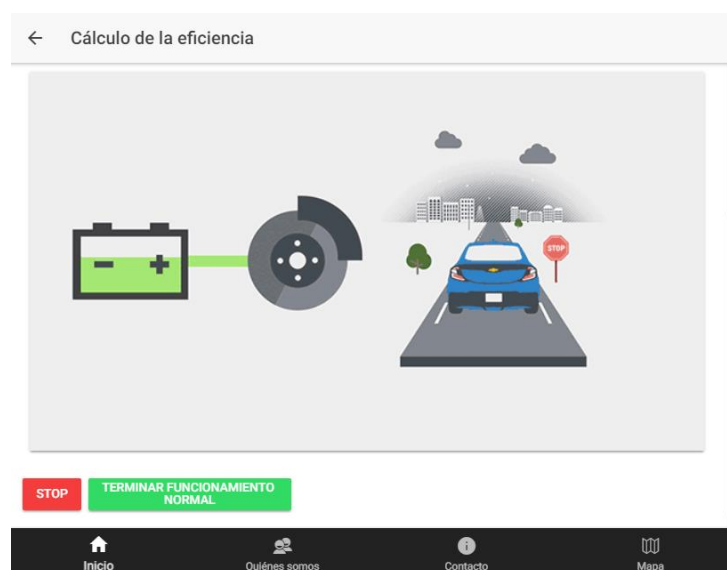


Figura 30: apariencia de la página donde se calcula la eficiencia en el funcionamiento normal (parte 2).

- **Página “Quiénes somos”**

En “Quiénes somos”, se explica quién es la autora de este proyecto, por qué surge y los beneficios que se obtendrán con esta aplicación.

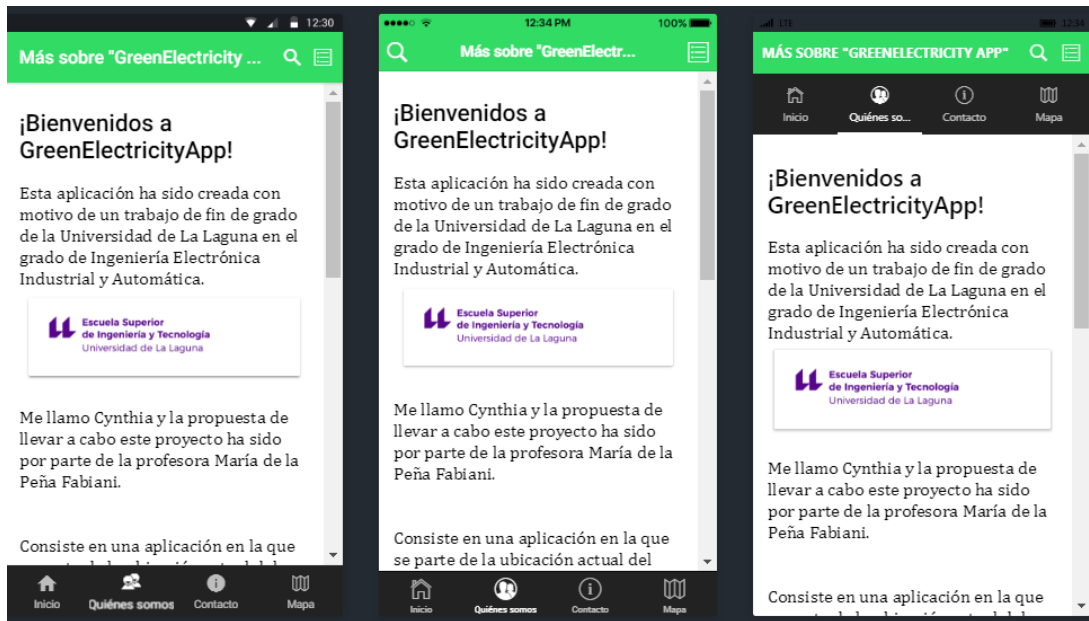


Figura 31: apariencia de la página de “Quiénes somos” en Android, iOS y Windows respectivamente (parte1).

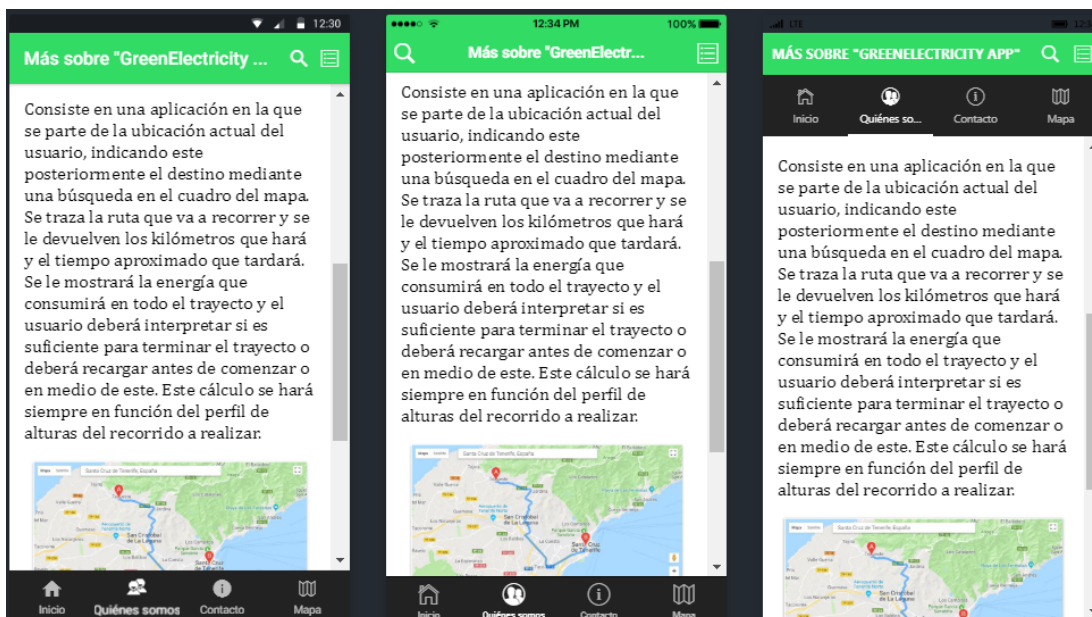


Figura 32: apariencia de la página de “Quiénes somos” en Android, iOS y Windows respectivamente (parte2).

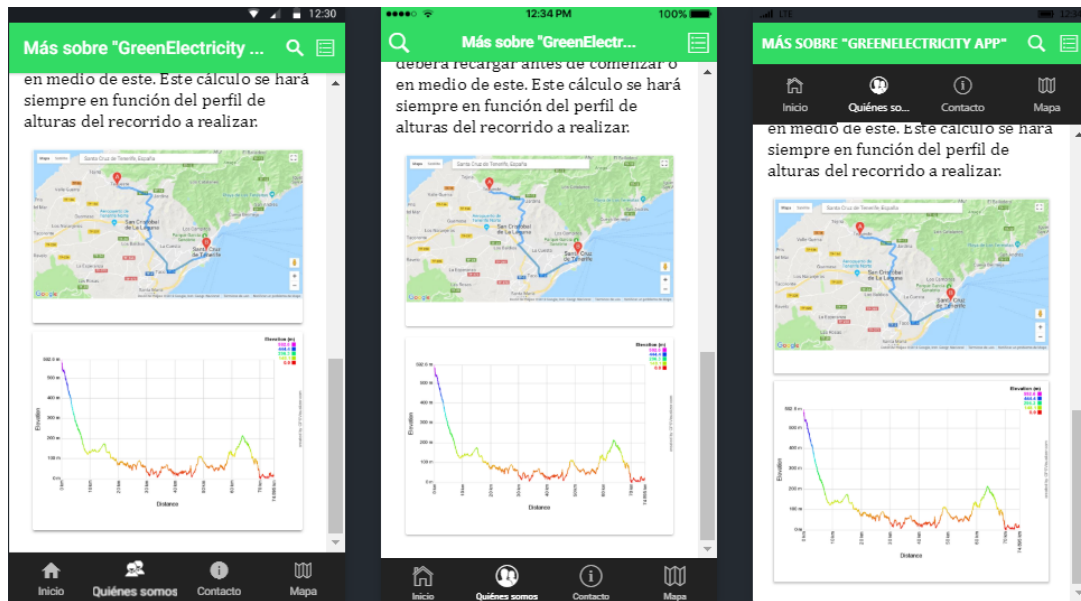


Figura 33: apariencia de la página de “Quiénes somos” en Android, iOS y Windows respectivamente (parte3).

- **Página “Contacto”**

En “Contacto” se da la oportunidad de acceder a la Universidad de La Laguna y al repositorio de esta universidad para poder ver otros trabajos de fin de grado.

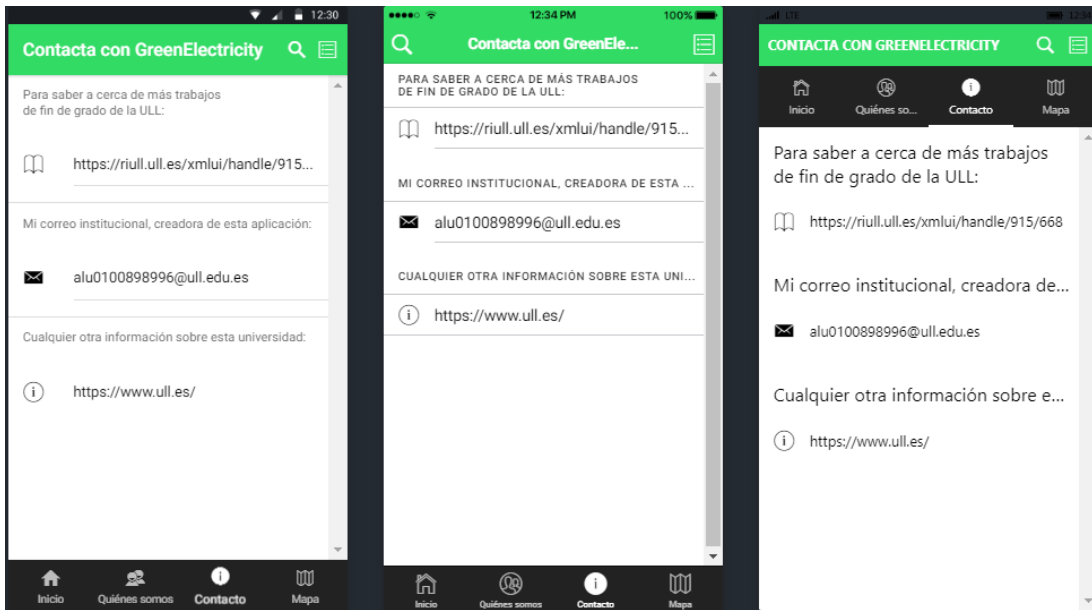


Figura 34: apariencia de la página de “Contacto” en Android, iOS y Windows respectivamente.

- **Página “Mapa”**

En esta página se visualizará el mapa con la ubicación actual del usuario.

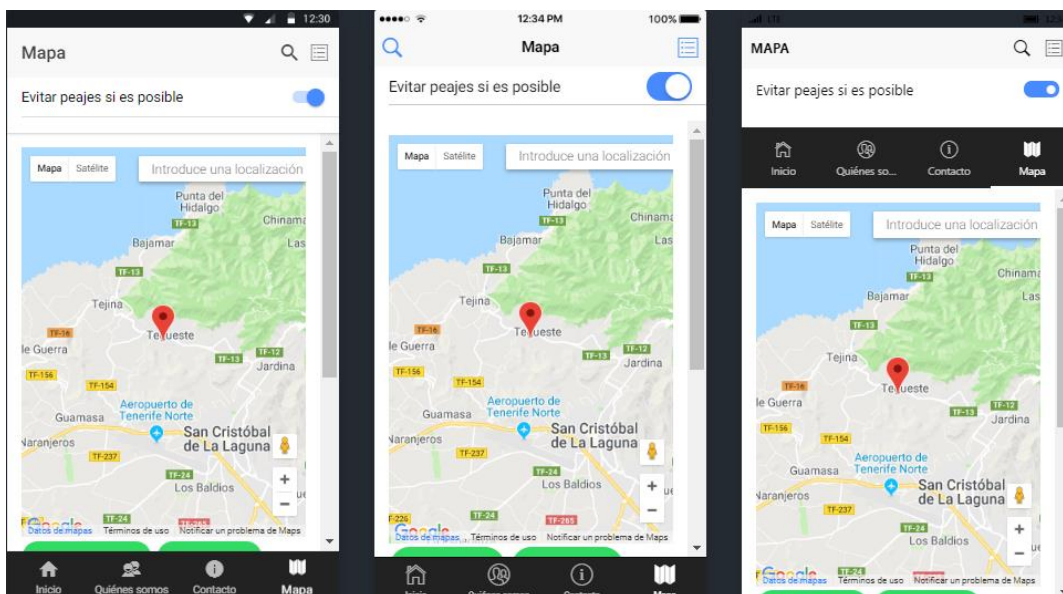


Figura 35: apariencia de la página de “Mapa” en Android, iOS y Windows respectivamente (parte I).

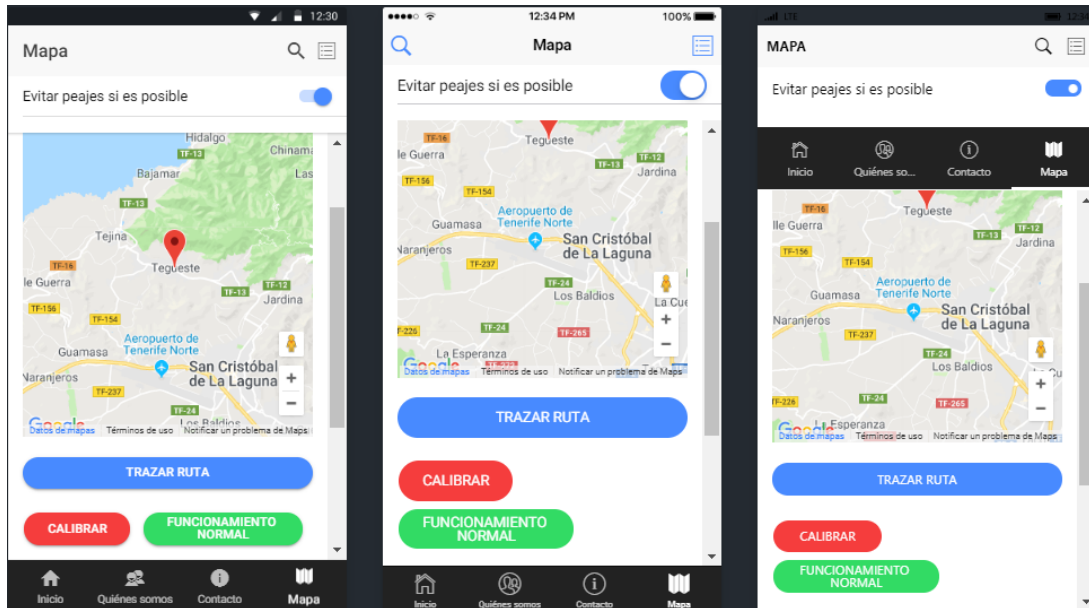


Figura 36: apariencia de la página de “Mapa” en Android, iOS y Windows respectivamente (parte2).

Al pie de esta página se encuentran tres botones: “Trazar ruta” es para trazar la ruta una vez se indica el punto de destino, y “Calibrar” y “Funcionamiento normal” serán los botones a los que habrá que darle para realizar los cálculos, dependiendo del modo que se esté ejecutando.

11.6.2. Menú principal

En la esquina superior derecha de estas cuatro páginas se muestra el icono de menú. Al hacer clic sobre él, se abrirá una nueva página en donde se mostrará una lista de opciones. Además, en cada una de las opciones de este menú, se ha incluido una flecha en la esquina superior izquierda, para volver a atrás:



Figura 37: vista desde el menú en la parte superior derecha desde una de las páginas del menú horizontal.



Figura 38: estructura del menú.

- **Iniciar sesión:** al pinchar sobre este botón se abrirá una página en la que aparecerá un cuadro para introducir el correo electrónico y otro para la contraseña, además de un botón para acceder.

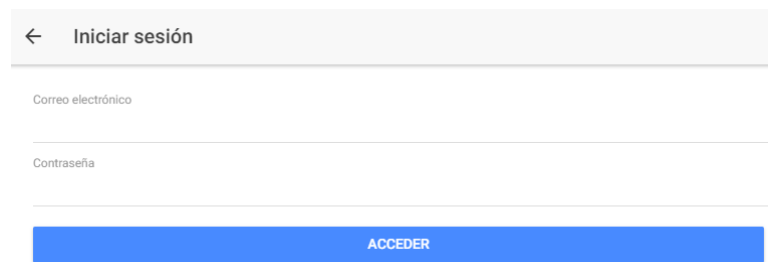


Figura 39: apariencia de la página que se abre al darle a iniciar sesión.

- **Registrarse:** sigue la misma estructura que el apartado de iniciar sesión, pero el botón será para registrarse como usuario en la aplicación en este caso.

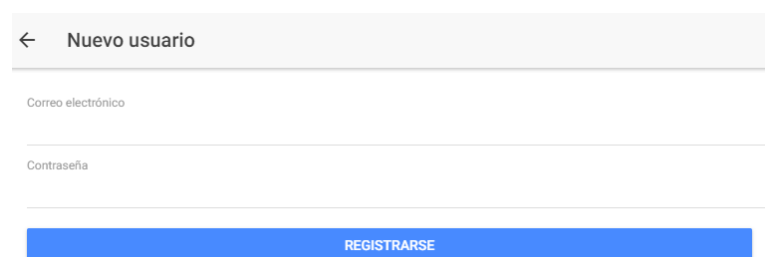


Figura 40: apariencia de la página que se abre al darle a nuevo usuario.

- **Comenzar nueva ruta:** Al darle a este botón, la aplicación redirigirá al usuario hasta la página en donde elegirá destino nuevamente para calcular una nueva ruta.

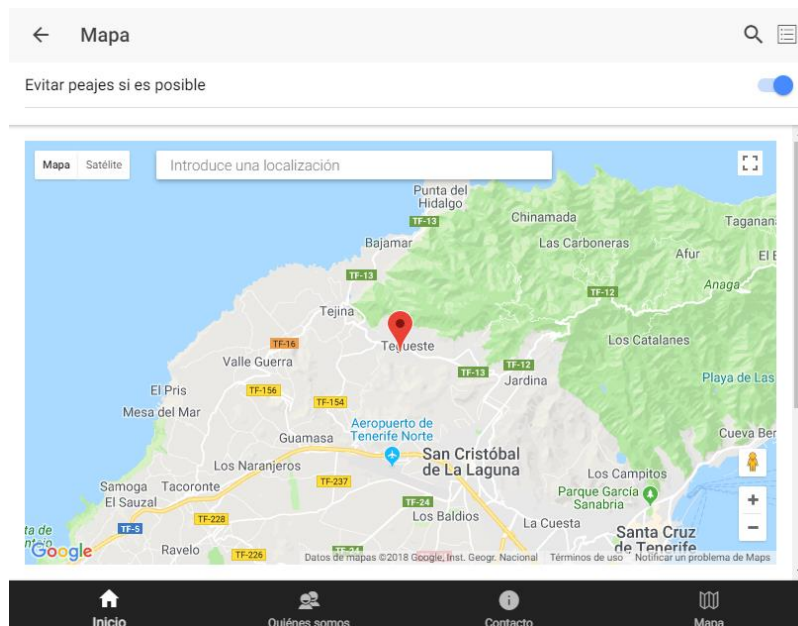


Figura 41: página a la que se redirige al pulsar el botón de nueva ruta.

- **Cerrar:** se cerrará la sesión del usuario volviendo nuevamente a la página de inicio.

12. Posibles mejoras en la aplicación

Uno de los principales retos encontrados al comienzo de este proyecto ha sido la falta de conocimientos previos en programación. Ello, junto al tiempo limitado de que se dispone en un trabajo de fin de grado, nos ha impedido realizar todas las mejoras planteadas en el proceso de desarrollo. Estas son:

- Mejora en la estimación de la velocidad en cada tramo para el modo de funcionamiento normal: en el modo de calibración, la aplicación calcula de forma automática la velocidad en todo momento, pero en el modo de funcionamiento predictivo, dicha cantidad ha de ser estimada. Actualmente está programado para que el usuario introduzca como *input* la velocidad que cree que va a tomar en el recorrido. No es una solución muy óptima puesto que si estamos mejorando la predicción de autonomía con la altura pero no la mejoramos todo lo que se podría en su dependencia con las velocidades, no se obtiene una solución todo lo real que sería deseable. Aun así, sigue siendo más real que la proporcionada por las aplicaciones de los vehículos eléctricos, basada en las velocidades medias durante los últimos kilómetros recorridos (es habitual, por ejemplo, que tras un prolongado uso del vehículo en ciudad la autonomía por autopista se infravalore considerablemente). La mejora que proponemos consistiría en, una vez trazada la ruta que va a recorrer el usuario, averiguar qué tipo de carreteras va a recorrer y, dependiendo de eso, establecer como velocidad supuesta la máxima permitida o algún valor estadístico de los proporcionados por Google.
- Estimación del consumo en climatización: una de las causas de la disminución de autonomía de los vehículos eléctricos es el uso de sistemas de refrigeración/calefacción. Este factor no se puede tener en cuenta con la estimación convencional de autonomía y sería relativamente sencillo introducirlo en nuestra aplicación, a partir de los datos de consumo del sistema y de los datos meteorológicos de temperatura.
- Mejora de acceso a la aplicación: otra mejora sería que el usuario pudiera registrarse en la aplicación por medio de otras plataformas, como por ejemplo Gmail o Facebook.

- Mejoras en las representaciones gráficas: sería interesante, por ejemplo, incorporar una gráfica con el perfil de alturas del trayecto que se va a trazar, a través de Google Earth.

13. Conclusions

This project is posed like a solution to the problem that exists with the vehicle electric apps, which calculate the autonomy from the speed in the last kilometres.

After doing the energetic study of the electric vehicle, it has been obtained as a conclusion that, given the high of the journey, the autonomy that results is much more accurate, a fact that attracts the user so much. This is a feature that is very interesting in the buyers of this kind of vehicles because they search that the vehicle has got the same autonomy than the fuel vehicles. Give an autonomy more accurate than the own app of the electric vehicle gives, is a solution very attractive to the consumers.

Also, the result has been an app very visual and interactive, that are features very important when it throws a mobile app.

The conclusions about Ionic are that it is an intuitive system, that uses TypeScript instead of JavaScript and it helps the programming. Its compilation in a web navigator makes more representative the code and the detection of fails at time that it is written, is an essential feature for beginner programmers like me.

14. Anexos

14.1. Anexo I: tablas con las características de los vehículos de este proyecto

Renault ZOE	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	26
Ancho (mm)	1.730
Alto (mm)	1.562
Peso (kg)	1470
Coefficiente aerodinámico (C_x)	0.34
Autonomía (km)	240

Tabla 3: datos del Renault ZOE

Smart ForTwo eD Cabrio, Coupé	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	17.6
Ancho (mm)	1.559
Alto (mm)	1.565
Peso (kg)	975
Coefficiente aerodinámico (C_x)	0.29
Autonomía (km)	145

Tabla 4: datos del Smart ForTwo eD Cabrio, Coupé

Kangoo Renault ZE	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio

Capacidad de batería (kWh)	33
Ancho (mm)	1.829
Alto (mm)	1.818
Peso (kg)	1410
Coefficiente aerodinámico (C_x)	0.33
Autonomía (km)	160

Tabla 5: datos del Renault Kangoo Renault ZE

Nissan LEAF Acenta, Tekna, Visia, Visia+	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	30
Ancho (mm)	1.770
Alto (mm)	1.550
Peso (kg)	1610
Coefficiente aerodinámico (C_x)	0.29
Autonomía (km)	250

Tabla 6: datos del Nissan LEAF Acenta, Tekna, Visia, Visia+

Volkswagen e-Golf 100 kW	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	35.8
Ancho (mm)	1.799
Alto (mm)	1.482
Peso (kg)	1615
Coefficiente aerodinámico (C_x)	0.28
Autonomía (km)	250

Tabla 7: datos del Volkswagen e-Golf 100kW

Volkswagen e-Golf 85 kW	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	24.2
Ancho (mm)	1.799
Alto (mm)	1.453
Peso (kg)	1585
Coefficiente aerodinámico (C_x)	0.28
Autonomía (km)	250

Tabla 8: datos del Volkswagen e-Golf 85 kW

BMW i3	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	22
Ancho (mm)	1.775
Alto (mm)	1.578
Peso (kg)	1280
Coefficiente aerodinámico (C_x)	0.29
Autonomía (km)	130

Tabla 9: datos del BMW i3

Hyundai IONIQ HE	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	28
Ancho (mm)	1.475
Alto (mm)	1.610
Peso (kg)	1110
Coefficiente aerodinámico (C_x)	0.24
Autonomía (km)	250

Tabla 10: datos del Hyundai IONIQ HE

Peugeot iON	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	14.5
Ancho (mm)	1.475
Alto (mm)	1.608
Peso (kg)	1195
Coefficiente aerodinámico (C_x)	0.33
Autonomía (km)	150

Tabla 11: datos del Peugeot iON

Kia SOUL EV	
Motor	Síncrono de imanes permanentes
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	27
Ancho (mm)	1.800
Alto (mm)	1.593
Peso (kg)	1565
Coefficiente aerodinámico (C_x)	0.33
Autonomía (km)	145

Tabla 12: datos del Kia SOUL EV

Tesla Model S 75D	
Motor	Asíncrono de rotor bobinado
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	75
Ancho (mm)	1.964
Alto (mm)	1.445
Peso (kg)	2183
Coefficiente aerodinámico (C_x)	0.24

Autonomía (km)	490
-----------------------	-----

Tabla 13: datos del Tesla Model S 75D

Tesla Model S P100D	
Motor	Asíncrono de rotor bobinado
Tipo de batería	Ion-Litio
Capacidad de batería (kWh)	100
Ancho (mm)	1.964
Alto (mm)	1.445
Peso (kg)	2316
Coefficiente aerodinámico (C_x)	0.24
Autonomía (km)	613

Tabla 14: datos del Tesla Model S P100D

14.2. Anexo II: Código de la programación de la aplicación

- Página “Home”
 - Home.html

```

<ion-header>
  <ion-navbar color="secondary">
    <ion-buttons end>
      <button ion-button icon-only color="dark"
(click)="iramenu()">
        <ion-icon name="ios-list-box-outline"></ion-icon>
      </button>
    </ion-buttons>
  <ion-title color="dark">Inicio</ion-title>
  <ion-buttons start>

    <button ion-button icon-only color="dark">
      <ion-icon name="ios-contact"></ion-icon>
    </button>
  </ion-buttons>
</ion-header>

```

```

        <button ion-button icon-only color = "dark">
            <ion-icon name="search"></ion-icon>
        </button>
    </ion-buttons>
</ion-navbar>
</ion-header>

<ion-content padding>
    <ion-card>
        
        <ion-card-content>

            </ion-card-content>
        </ion-card>
        <p id ="parrafo">
            ¡Conoce los kilómetros de autonomía que le queda a tu vehículo
            según el perfil de alturas que se recorra! </p>
        <p id ="parrafo1">
            &nbsp; <!--Espacio-->
            &nbsp;
            &nbsp;
            ¡Comencemos!
            <br> <!--Salto de línea-->
            <br>
            <ion-card> <!--Insertar imagen-->
                
            </ion-card>
        </p>

        <button ion-button block color="danger"
        (click)="calibrar()">CALIBRACIÓN</button>
        <button ion-button block color="secondary"
        (click)="InformacionVehiculo()">FUNCIONAMIENTO NORMAL</button>
        <!--Al hacer clic sobre el botón, se ejecutará el método
        InformacionVehiculo()-->

        <button ion-button block (click)="registrarse()">NUEVO
        USUARIO</button>

</ion-content>

```

- Home.ts

```

import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';

```

```

import { InforVehiculoPage } from '../infor-vehiculo/infor-vehiculo';
import { MenuPage } from '../menu/menu';
import { CalibracionPage } from '../calibracion/calibracion';
import { RegistroPage } from '../registro/registro';

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {

  boton_usuario : any;
  menupage : any;

  constructor(public navCtrl: NavController) {
    this.menupage= MenuPage; //Nuevo nombre de la clase para poder
    importar variables.
  }
  InformacionVehiculo() {
    this.navCtrl.push(InforVehiculoPage); //Mediante navCtrl se
    permite la navegación entre páginas.
  }
  iramenu(){
    this.navCtrl.push(this.menupage,{
      usuario: this.boton_usuario //Variable exportada
    });
  }
  calibrar(){
    this.navCtrl.push(CalibracionPage);
  }

  registrarse (){
    this.navCtrl.push(RegistroPage);
  }
}

```

- Home.css

```

page-home {

  ion-content{

```

```

    #parrafo { //Se indica la fuente de la letra del párrafo, el
tamaño y el grosor.
    font-family: Cambria, Cochin, Georgia, Times, 'Times New
Roman', serif;
    font-size: 20px;
    font-weight: 300;
    }

    #parrafo1 {
        font-family: Cambria, Cochin, Georgia, Times, 'Times New
Roman', serif, ;
        font-size: 30px;
        font-weight: 700; //Negrita
        color: red;
    }
}
}

```

- Contact.html

```

<ion-header>
  <ion-navbar color="secondary">
    <ion-buttons end>
      <button ion-button icon-only (click)="iramenucontact()">
<!--Botón de solo icono-->
        <ion-icon name="ios-list-box-outline"></ion-icon>
      </button>
    </ion-buttons>
    <ion-title>
      Contacta con GreenElectricity
    </ion-title>
    <ion-buttons start>
      <button ion-button icon-only>
        <ion-icon name="search"></ion-icon>
      </button>
    </ion-buttons>
  </ion-navbar>
</ion-header>

<ion-content>
  <ion-list>
    <ion-list-header>Para saber a cerca de más trabajos <br> de
fin de grado de la ULL:</ion-list-header>
    <ion-item>
      <ion-icon name="ios-book-outline" item-start></ion-icon>

```

```

    https://riull.ull.es/xmlui/handle/915/668
  </ion-item>
<br>
  <ion-list-header>Mi correo institucional, creadora de esta
aplicación:</ion-list-header>
  <ion-item>
    <ion-icon name="ios-mail" item-start></ion-icon>
    alu0100898996@ull.edu.es
  </ion-item>
  <br>
  <ion-list-header>Cualquier otra información sobre esta
universidad:</ion-list-header>
  <ion-item>
    <ion-icon name="ios-information-circle-outline" item-
start></ion-icon>
    https://www.ull.es/
  </ion-item>
</ion-list>
</ion-content>

```

- Contact.ts

```

import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { MenuPage } from '../menu/menu';

@Component({
  selector: 'page-contact',
  templateUrl: 'contact.html'
})
export class ContactPage {

  constructor(public navCtrl: NavController) {

  }

  iramenucontact(){
    this.navCtrl.push(MenuPage); //push es el método
  }

}

```

- About.html

```

<ion-header>
  <ion-navbar color="secondary">
    <ion-buttons end>
      <button ion-button icon-only (click)="iramenuabout()">

```

```

        <ion-icon name="ios-list-box-outline"></ion-icon>
      </button>
    </ion-buttons>
  <ion-title>
    Más sobre "GreenElectricity App"
  </ion-title>
  <ion-buttons start>
    <button ion-button icon-only>
      <ion-icon name="search"></ion-icon>
    </button>
  </ion-buttons>
</ion-navbar>
</ion-header>

<ion-content padding>
  &nbsp;
  &nbsp;
  <h1>¡Bienvenidos a GreenElectricityApp!</h1>

  <p id="parrafo6">
    Esta aplicación ha sido creada con motivo de un trabajo de
    fin de grado de la Universidad de
    La Laguna en el grado de Ingeniería Electrónica Industrial y
    Automática.
    <ion-card> <!--Insertar imagen-->
      
    </ion-card>
    <br>
    Me llamo Cynthia y la propuesta de llevar a cabo este
    proyecto ha sido por parte
    de la profesora María de la Peña Fabiani.
    <br> <br> <br>
    Consiste en una aplicación en la que se parte de la
    ubicación actual del usuario, indicando
    este posteriormente el destino mediante una búsqueda en el
    cuadro del mapa.
    Se traza la ruta que va a recorrer y se le devuelven los
    kilómetros que hará
    y el tiempo aproximado que tardará. Se le mostrará la
    energía que consumirá en
    todo el trayecto y el usuario deberá interpretar si es
    suficiente para
    terminar el trayecto o deberá recargar antes de comenzar o
    en medio de este. Este cálculo se
    hará siempre en función del perfil de alturas del recorrido
    a realizar.
  </p>
  <ion-card>

```



```

    
  </ion-card>
  <ion-card>
    

  </ion-card>

</ion-content>

```

- About.ts

```

import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { MenuPage } from '../menu/menu';

@Component({
  selector: 'page-about',
  templateUrl: 'about.html'
})
export class AboutPage {

  constructor(public navCtrl: NavController) {
  }

  iramenuabout(){
    this.navCtrl.push(MenuPage);
  }
}

```

- Calculocorto.html

```

<ion-header>

  <ion-navbar>
    <ion-title>Calibración en un tramo corto</ion-title>
  </ion-navbar>

</ion-header>

<ion-content padding>

  <ion-item>
    <ion-label>Superficie (m2):</ion-label>
    <ion-input type="number" [(ngModel)]="demo2" id="demo1"
disabled></ion-input> //Input de números deshabilitado (no se puede

```

modificar su valor). Se utiliza para mostrar el resultado de la superficie total.

```

</ion-item>
<ion-item>
  <ion-label>Peso total (kg):</ion-label>
  <ion-input type="number" [(ngModel)]="peso2"
id="peso_total1" disabled></ion-input> // ngModel le da un
identificador a los inputs, de manera que se declara este valor en
la clase del .ts y se puede acceder al valor del input, pudiendo
modificarlo.
</ion-item>

<ion-item>
  <ion-label>HORAS:</ion-label>
  <ion-input type="number" value="00" [(ngModel)]="horas1"
id="horas1" disabled></ion-input>
</ion-item>
<hr> <!--línea que separa-->
<ion-item>
  <ion-label>MINUTOS:</ion-label>
  <ion-input type="number" value="00" [(ngModel)]="minutos1"
id="minutos1" disabled></ion-input>
</ion-item>

<ion-item>
  <ion-label>SEGUNDOS:</ion-label>
  <ion-input type="number" value="00"
[(ngModel)]="segundos1" id="segundos1" disabled></ion-input>
</ion-item>

<ion-card>
  

</ion-card>

<br>
<button id="boton_stop" ion-button color="danger" onclick
="cronoOff()" STOP </button>
<button id="bot_calibrar" ion-button color="secondary" (click)
="terminarcalibracion()"> Terminar calibración </button>

<ion-item>
  <ion-label>Energía consumida durante el trayecto:</ion-
label>
  <ion-input type="number" [(ngModel)]="energiaC"
id="energiaC" disabled></ion-input>

```

```

    </ion-item>

    <ion-item>
      <ion-label>Energía que se ha tenido que recargar para <br>
consumir dicha energía: </ion-label>
      <ion-input type="number" [(ngModel)]="energiaR" id="energiaR"
disabled></ion-input>
    </ion-item>
  </ion-content>

```

- o Calculocorto.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { InforVehiculoPage } from '../infor-vehiculo/infor-
vehiculo';

@IonicPage()
@Component({
  selector: 'page-calculo-corto',
  templateUrl: 'calculo-corto.html',
})
export class CalculoCortoPage {

  //Variables declaradas dentro de la clase pero fuera del
constructor. Serán de tipo any y globales. Se accede a ellas por
medio de la palabra reservada this.

  inforvehiculoPage : any;

  anchoC2 : any;
  altoC2 : any;
  personasC2 : any;
  pesoC2 : any;
  capacidadC2 : any;
  intervaloC2 : any;
  cdC2 : any;
  nombre2 : any;
  altura2 : any;
  distancia2 : any;

  superficie2 : any;
  peso2 : any;
  horas2 : any;
  minutos2 : any;
  segundos2 : any;

```

```
tiempo : any;
cont2 : any;
chrono2 : any;
energiaC : any;
energiaR : any;

constructor(public navCtrl: NavController, public navParams:
NavParams) {

    this.inforvehiculoPage = InforVehiculoPage;
//Calibración (importación de variables)
    this.anchoC2=navParams.data.anchoC2;
    this.altoC2=navParams.data.altoC2;
    this.personasC2=navParams.data.personasC2;
    this.pesoC2=navParams.data.pesoC2;
    this.capacidadC2=navParams.data.capacidadC2;
    this.intervaloC2=navParams.data.intervaloC2;
    this.cdC2=navParams.data.cdC2;
    this.nombre2=navParams.data.nombre2;
    this.altura2=navParams.data.altura2;
    this.distancia2=navParams.data.distancia2;

//Todas las variables que se utilicen en las funciones se declararán
en el constructor.

    var x = this.anchoC2; //x e y variables actuales
    var y = this.altoC2;

    this.superficie2 = this.sup (x,y); //Invocación del método sup.

    var a = this.personasC2;
    var b = this.pesoC2;

    this.peso2 = this.pes (a,b); //Invocación del método pes.

    var crr = 0.03; //Fijado según las condiciones del terreno.
    var distancia = 1; //Intervalo fijo de 1 km.
    var ef_reg = 0.8; //Eficiencia regenerativa.
    var velocidad=distancia/(this.tiempo/60);
    var superficie=this.superficie2;
    var ef_global = 0.85; //Eficiencia global.
    var ef_carga = 0.8; //Eficiencia de carga.

    this.calibracion (crr, distancia, ef_reg, velocidad, superficie,
ef_global, ef_carga);
```

```
this.temp();
}

ionViewDidLoad() {
  console.log('ionViewDidLoad CalculoCortoPage');
}

terminarcalibracion(){
  this.navCtrl.push(this.inforvehiculoPage, {
    ancho3 : this.anchoC2,
    alto3: this.altoC2,
    capacidad3 : this.capacidadC2,
    personas3 : this.personasC2,
    peso3 : this.pesoC2,
    intervalo3 : this.intervaloC2,
    cd3 : this.cdC2,
    nombre3 : this.nombre2,
    distancia3 : this.distancia2
  }) //Variables exportadas.
}

sup (an, al) { //an y al, variables locales.
  var sup_frontal = an * al;
  return sup_frontal;
}

pes (per, pes) {
  var peso_total = (per*80) + pes;
  return peso_total;
}

temp() {
  var contador_s=0;
  var contador_m=0;
  var contador_h=0;

  this.chrono2 = setInterval(function() { //setInterval, función
que inicia el crono.

    if (contador_s == 60){
      contador_s = 0;
      contador_m++;
      this.minutos = contador_m;
      if (contador_m == 60){
```

```

        contador_m=0;
        contador_h++;
        this.horas = contador_h;
        if(contador_h == 24){
            contador_h=0;
        }
    }
}

this.segundos= contador_s;
++contador_s;
console.log(this.segundos);

this.tiempo = contador_s + (contador_m*60) +
(contador_h*1200); //Tiempo en cada momento en segundos.

},1000);
}

cronoOff(){
    clearInterval(this.chrono); //Para el crono.
}

calibracion (crr1, distancia1, ef_reg1, velocidad1, superficie1,
ef_global1, ef_carga1) {

//distancia2 es la distancia total a recorrer.

    for (let i=1; i <= this.distancia2.options.length; i++) {

        let distancia_inicial = this.distancia2.options.length;

        if (distancia_inicial = this.distancia2.length - 1) {
//Cuando haya recorrido 1 km de distancia

            var Ec=((-
0.5*this.peso2*((velocidad1*1000/3600)^2))/1000/3600); //Energía
cinética.
            var Ep=(-this.peso2*9.81*this.altura2/1000/3600)*ef_reg1;
//Energía potencial.

```

```

    var Eaero= (-
0.5*this.cdC2*superficie1*1.2*(velocidad1*1000/3600)^2*distancia1*10
00/1000/3600); //Energía aerodinámica.
    var Erod= (-crr1*this.peso2*9.81*distancia1*1000/1000/3600);
//Energía de rodadura.

    }
}

    var Eacc = 1.5 * (this.tiempo /60); //Pérdidas en los accesorios
del vehículo.
    this.energiaC = ((Ec + Ep + Eaero + Erod)/ef_global1) + Eacc;
//Pérdidas de energía durante el recorrido.
    this.energiaR = this.energiaC/ef_carga1; //Energía de recarga
que habrá que acumular para poder consumir la que se pierde.

    return this.energiaC, this.energiaR;

};

}

```

- o Calculoeficiencia.html

```

<ion-content padding>
  <ion-item>
    <ion-label>Superficie (m2):</ion-label>
    <ion-input type="number" [(ngModel)]="demo" id="demo"
disabled></ion-input>
  </ion-item>
  <ion-item>
    <ion-label>Peso total (kg):</ion-label>
    <ion-input type="number" [(ngModel)]="peso" id="peso_total"
disabled></ion-input>
  </ion-item>

  <ion-item>
    <ion-label>HORAS:</ion-label>
    <ion-input type="number" value="00" [(ngModel)]="horas"
id="horas" disabled></ion-input>

```

```

</ion-item>

<ion-item>
  <ion-label>MINUTOS:</ion-label>
  <ion-input type="number" value="00" [(ngModel)]="minutos" id
="minutos" disabled></ion-input>
</ion-item>

<ion-item>
  <ion-label>SEGUNDOS:</ion-label>
  <ion-input type="number" value="03" [(ngModel)]="segundos"
id="segundos" disabled></ion-input>
</ion-item>

<ion-card>
  

</ion-card>

<br>

  <ion-item>
    <ion-label>Para hacer este trayecto, el <br> vehículo gastará:
</ion-label>
    <ion-input type="number" [(ngModel)]="energiaG"
disabled></ion-input>
  </ion-item>

  <ion-item>
    <ion-label>Y habrá que recargar en la batería:</ion-label>
    <ion-input type="number" [(ngModel)]="energiaR" disabled></ion-
input>
  </ion-item>

</ion-content>
<button id="stop_calib" ion-button color="secondary" onclick
="eficiencia()"> COMENZAR A <br> FUNCIONAR </button>

<br> <!--Salto de página-->
<br>

<button id="boton_stop" ion-button color="danger" onclick
="cronoOff()"> STOP </button>

  <button id="stop_calib" ion-button color="secondary" onclick
="gotoinicio()"> TERMINAR FUNCIONAMIENTO <BR> NORMAL </button>

```

- Calculoeficiencia.ts


```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { TabsPage } from '../tabs/tabs';

@IonicPage()
@Component({
  selector: 'page-calculo-eficiencia',
  templateUrl: 'calculo-eficiencia.html',
})
export class CalculoEficienciaPage {

  personas5 : any;
  crr5 : any;
  cd5 : any;

  ancho5 : any;
  alto5 : any;
  peso5 : any;
  capacidad5 : any;
  intervalo5 : any;
  distancia5 : any;
  altura5 : any;
  veloc5 : any;

  superficie3 : any;
  peso : any;
  horas : any;
  minutos : any;
  segundos : any;
  cont : any;
  chrono : any;
  tiempo : any;

  energiaG : any;
  energiaR : any;

  //Si se pone en el constructor, se ejecutará nada más entrar en la
  //página, por orden de declaración

  constructor(public navCtrl: NavController, public NavParams:
  NavParams) {

    this.personas5=navParams.data.personas5;
```

```
this.crr5=navParams.data.crr5;
this.cd5=navParams.data.cd5;

//Parámetros de la calibración al cálculo del funcionamiento normal
this.ancho5=navParams.data.ancho5;
this.alto5=navParams.data.alto5;
this.peso5=navParams.data.peso5;
this.capacidad5=navParams.data.capacidad5;
this.intervalo5=navParams.data.intervalo5;
this.distancia5=navParams.data.distancia5;
this.altura5=navParams.data.altura5;
this.veloc5=navParams.data.veloc5;

console.log(this.ancho5);
console.log(this.alto5);
console.log(this.capacidad5);
console.log(this.personas5);
console.log(this.cd5);
console.log(this.crr5);
console.log(this.peso5);
console.log(this.intervalo5);

var x = this.ancho5; //variables locales
var y = this.alto5;

this.superficie3 = this.sup (x,y); //x y variables actuales

var a = this.personas5;
var b = this.peso5;

this.peso = this.pes (a,b);

var crr = 0.03; //Fijado según las condiciones del terreno.
var distancia = 1; //Intervalo fijo de 1 km.
var ef_reg = 0.8;
var velocidad=distancia/(this.tiempo/60);
var superficie=this.superficie3;
var ef_global = 0.85;
var ef_carga = 0.8;

this.eficiencia (crr, distancia, ef_reg, velocidad, superficie,
ef_global, ef_carga);
```

```
}

ionViewDidLoad() {
    console.log('ionViewDidLoad CalculoEficienciaPage');
}

gotoinicio(){
    this.navCtrl.push(TabsPage);
}

sup (an, al) {
    var sup_frontal = an * al;
    return sup_frontal;
}

pes (per, pes) {
    var peso_total = (per*80) + pes;
    return peso_total;
}

cronoOff(){
    clearInterval(this.chrono); //Para el crono.
}

eficiencia (crr1, distancia1, ef_reg1, velocidad1, superficie1,
ef_global1, ef_carga1) { //Hace uso del crono

    var contador_s=0;
    var contador_m=0;
    var contador_h=0;
    this.chrono = setInterval(function() {

        if (contador_s == 60){
            contador_s = 0;
            contador_m++;
            this.minutos = contador_m;
```

```

    if (contador_m == 60){
      contador_m=0;
      contador_h++;
      this.horas = contador_h;
      if(contador_h == 24){
        contador_h=0;
      }
    }
  }

  this.segundos= contador_s;
  ++contador_s;
  console.log(this.segundos);

  this.tiempo = contador_s + (contador_m*60) + (contador_h*1200);
},1000);

//Distancia5 es la distancia total a recorrer.

for (let i=1; i <= this.distancia5.options.length; i++) {
//Cuando haya recorrido 1 km de distancia

  let dis_inicial = this.distancia5.options.length;

  if (this.distancia5.length = dis_inicial - 1) {

    var Ec=((-
0.5*this.peso*((velocidad1*1000/3600)^2))/1000/3600); //Energía
cinética.
    var Ep=(-
this.peso*9.81*this.altura5/1000/3600)*ef_reg1;//Energía potencial.
    var Eaero= (-
0.5*this.cd5*superficie1*1.2*(velocidad1*1000/3600)^2*distancia1*100
0/1000/3600); //Energía aerodinámica.
    var Erod= (-crr1*this.peso*9.81*distancia1*1000/1000/3600);
//Energía de rodadura.

  }

}

var Eacc = 1.5 * (this.tiempo /60); //Pérdidas en los accesorios.
this.energiaG = ((Ec + Ep + Eaero + Erod)/ef_global1) + Eacc;
//Pérdidas totals en el trayecto.

```

```
    this.energiaR = this.energiaG/ef_carga1; //Energía que habrá que
recargar para poder consumir la indicada.

    return this.energiaG, this.energiaR, this.tiempo;
};

prediccion (crr2, distancia2, ef_reg2, velocidad2, superficie2,
ef_global2, ef_carga2) {

    this.veloc5 = velocidad2; //Velocidad media que se introduce de
input.

    for (let i=1; i <= this.distancia5.options.length; i++) {

        let dis_inicial = this.distancia5.options.length;

        if (this.distancia5.length = dis_inicial - 1) {

            var Ec=((-
0.5*this.peso*((velocidad2*1000/3600)^2))/1000/3600); //Energía
cinética.
            var Ep=(-this.peso*9.81*this.altura5/1000/3600)*ef_reg2;
//Energía potencial.
            var Eaero= (-
0.5*this.cd5*superficie2*1.2*(velocidad2*1000/3600)^2*distancia2*100
0/1000/3600); //Distancia2= 1 km. Energía aerodinámica.

            var Erod= (-
crr2*this.peso*9.81*distancia2*1000/1000/3600); //Energía de
rodadura.

        }
    }

    var Eacc = 1.5 * (distancia2/velocidad2); //Pérdidas en los
accesorios.
    this.energiaG = ((Ec + Ep + Eaero + Erod)/ef_global2) + Eacc;
//Pérdidas totales de energía.
    this.energiaR = this.energiaG/ef_carga2; //Energía que habrá que
recargar para poder consumir la que se pierde.
}
```

}

- Calibracion.html

```

<ion-header>

  <ion-navbar>
    <ion-title>Calibrar</ion-title>
  </ion-navbar>

</ion-header>

<ion-content padding>
  <p id="parrafo2">La calibración es el paso inicial antes de
  comenzar a utilizar la aplicación en modo normal. Se trata de
  hacer un recorrido
  corto para ajustar los parámetros del vehículo lo máximo
  posible.
  </p>
  <p id="parrafo3"> -> Si selecciona uno de los modelos que
  se indican en la lista que se muestra dándole en el siguiente
  botón, se le adjudicarán algunos valores que vienen
  dados por su ficha técnica. <br>-> Si ud. cree que
  debe cambiarlos, podrá hacerlo. <br>-> Si su vehículo no se
  encuentra en la lista, seleccione "Otro".
  </p>

  <button ion-button color="danger" outline id=
  "boton_vehC" (click)="doRadio()">Modelo del vehículo</button>

  <p id="parrafo4"> Por favor, es muy importante que
  indique los valores en las unidades que se especifican.</p>

  <ion-list>

    <ion-item>
      <ion-label fixed>Ancho (m)</ion-label>
      <ion-input id= "dato_anchoC"
  [(ngModel)]="anchoC" type="number"></ion-input>
    </ion-item>

```

```

        <ion-item>
            <ion-label fixed>Alto (m)</ion-label>
            <ion-input id= "dato_altoC"
[(ngModel)]="altoC" type="number"></ion-input>
        </ion-item>

        <ion-item>
            <ion-label fixed>Cx (Cd)</ion-label>
            <ion-input id= "cdC" [(ngModel)]="cdC"
type="number"></ion-input>
        </ion-item>

<!--Se utiliza Popover para que se pueda elegir solo una
opción-->

        <ion-item>
            <ion-label>Número de pasajeros</ion-label>
            <ion-select [(ngModel)]="personasC"
id="num_pasajerosC" submitText="Ok" cancelText="Cancelar"
interface="popover">
<!--El valor que va en value es el que toma la propia
aplicación para hacer los cálculos y el que va entre las dos
etiquetas es lo que se ve en la aplicación-->
                <ion-option value="1">1</ion-option>
                <ion-option value="2">2</ion-option>
                <ion-option value="3">3</ion-option>
                <ion-option value="4">4</ion-option>
                <ion-option value="5">5</ion-option>
            </ion-select>
        </ion-item>

        <ion-item>
            <ion-label>Peso del vehículo (kg)</ion-label>
            <ion-input type="number"
[(ngModel)]="pesoC" id="peso_vehC"></ion-input>
        </ion-item>

    </ion-list>

    <button id="boton_okC" ion-button round block
color="secondary" (click)="Iralmapa()">ACEPTAR</button>

```

```

        <!--Al hacer clic sobre el botón, se ejecutará el
        método Iralmapa()-->

</ion-content>

```

- o Calibracion.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { AlertController } from 'ionic-angular';
import { MapaPage } from '../mapa/mapa';

@IonicPage()
@Component({
  selector: 'page-calibracion',
  templateUrl: 'calibracion.html',
})
export class CalibracionPage {

  testRadioOpen: boolean;
  testRadioResult;

  mapaPage : any;
  calculoeficienciaPage : any;

  anchoC : any;
  altoC : any;
  personasC : any;
  pesoC : any;
  capacidadC : any;
  intervaloC : any;
  cdC : any;

  nom : any;
  //NavController para navegar entre páginas y NavParams para
  importar variables entre páginas

  constructor(public navCtrl: NavController, public
  NavParams: NavParams, public alertCtrl: AlertController) {
    this.mapaPage=MapaPage;
  }

```



```
ionViewDidLoad() {  
  console.log('ionViewDidLoad CalibracionPage');  
}  
stpSelect() {  
  console.log('STP selected');  
}  
  
doRadio() { //Se crea una alerta para que se pueda elegir  
entre las opciones de una lista.  
  let alert = this.alertCtrl.create();  
  alert.setTitle('Vehículo');  
  
  alert.addInput({  
    type: 'radio',  
    label: 'Kia SOUL EV',  
    value: '1',  
    checked: true  
  });  
  
  alert.addInput({  
    type: 'radio',  
    label: 'Peugeot iON',  
    value: '2'  
  });  
  
  alert.addInput({  
    type: 'radio',  
    label: 'Renault ZE',  
    value: '3'  
  });  
  
  alert.addInput({  
    type: 'radio',  
    label: 'Hyundai IONIC HE',  
    value: '4'  
  });  
  
  alert.addInput({  
    type: 'radio',  
    label: 'Renault ZOE',  
    value: '5'  
  });  
  
  alert.addInput({  
    type: 'radio',  
    label: 'Nissan LEAF Acenta/Tekna/Visia/Visia+',  
    value: '6'  
  });  
}
```

```
alert.addInput({
  type: 'radio',
  label: 'BMW i3',
  value: '7'
});

alert.addInput({
  type: 'radio',
  label: 'Smart ForTwo eD Cabrio/Coupé', //Intentar que
salga en dos líneas para que se vea bien
  value: '8'
});

alert.addInput({
  type: 'radio',
  label: 'Tesla S 75D',
  value: '9'
});

alert.addInput({
  type: 'radio',
  label: 'Tesla S P100D',
  value: '10'
});

alert.addInput({
  type: 'radio',
  label: 'Volkswagen e-Golf 100 KW',
  value: '11'
});

alert.addInput({
  type: 'radio',
  label: 'Volkswagen e-Golf 85 KW',
  value: '12'
});

alert.addInput({
  type: 'radio',
  label: 'Otro',
  value: '13'
});

alert.addButton('Cancelar'); //Se añaden botones de
cancelar y aceptar.
alert.addButton({
  text: 'Ok',
  handler: data => {
    console.log('Radio data:', data);
  }
});
```

```
        this.testRadioOpen = false;
        this.testRadioResult = data; //data es el valor, es
decir, el número del vehículo elegido

//Dependiendo del vehículo escogido se tendrá un valor u otro
y, dependiendo de este, se colocarán automáticamente los
valores que correspondan obtenidos de la base de datos.
        if(this.testRadioResult == "1"){

            this.nom="Kia SOUL EV";

            this.anchoC = 1.8;
            this.altoC = 1.593;
            this.capacidadC = 27;
            this.pesoC = 1565;
            this.cdC = 0.33;
            document.getElementById("boton_vehC").innerHTML=
this.nom; //Sirve para copiar en el elemento con el id
correspondiente al que se coloca entre comillas
("botón_vehC"), el valor al que se iguala la expresión
(this.nom).

        }

        if(this.testRadioResult == "2"){

            this.nom="Peugeot iON";

            this.anchoC = 1.475;
            this.altoC = 1.608;
            this.capacidadC = 14.5;
            this.pesoC = 1195;
            this.cdC = 0.33;
            document.getElementById("boton_vehC").innerHTML =
this.nom;

        }

        if(this.testRadioResult == "3"){

            this.nom="Renault ZE";

            this.anchoC = 1.829;
            this.altoC = 1.818;
            this.capacidadC = 33;
            this.pesoC = 1410;
            this.cdC = 0.33;
            document.getElementById("boton_vehC").innerHTML =
this.nom;
```

```
    }

    if(this.testRadioResult == "4"){

        this.nom="Hyundai IONIQ BEV";

        this.anchoC = 1.475;
        this.altoC = 1.610;
        this.capacidadC = 28;
        this.pesoC = 1110;
        this.cdC = 0.24;
        document.getElementById("boton_vehC").innerHTML =
this.nom;

    }

    if(this.testRadioResult == "5"){

        this.nom="Renault ZOE";

        this.anchoC = 1.730;
        this.altoC = 1.562;
        this.capacidadC = 22;
        this.pesoC = 1470;
        this.cdC = 0.34;
        document.getElementById("boton_vehC").innerHTML =
this.nom;

    }

    if(this.testRadioResult == "6"){

        this.nom="Nissan LEAF Acenta/Tekna/Visia/Visia+";

        this.anchoC = 1.770;
        this.altoC = 1.550
        this.capacidadC = 30;
        this.pesoC = 1610;
        this.cdC = 0.29;
        document.getElementById("boton_vehC").innerHTML =
this.nom;

    }

    if(this.testRadioResult == "7"){
```

```
        this.nom="BMW i3";

        this.anchoC = 1.775;
        this.altoC = 1.578;
        this.capacidadC = 22;
        this.pesoC = 1280;
        this.cdC = 0.29;
        document.getElementById("boton_vehC").innerHTML =
this.nom;
    }

    if(this.testRadioResult == "8"){

        this.nom="Smart ForTwo eD Cabrio/Coupé";

        this.anchoC = 1.559;
        this.altoC = 1.565;
        this.capacidadC = 17.6;
        this.pesoC = 975;
        this.cdC = 0.29;
        document.getElementById("boton_vehC").innerHTML =
this.nom;
    }

    if(this.testRadioResult == "9"){

        //parseInt para que coja solo el número y no mm
        this.nom="Tesla S 75D";

        this.anchoC = 1.964;
        this.altoC= 1.445;
        this.capacidadC = 75;
        this.pesoC = 2183;
        this.cdC = 0.24;
        document.getElementById("boton_vehC").innerHTML =
this.nom;
    }

    if(this.testRadioResult == "10"){
        //parseInt para que coja solo el número y no mm
```

```
        this.nom="Tesla S P100D";

        this.anchoC = 1.964;
        this.altoC = 1.445;
        this.capacidadC = 100;
        this.pesoC = 2316;
        this.cdC = 0.24;
        document.getElementById("boton_vehC").innerHTML =
this.nom;

    }

    if(this.testRadioResult == "11"){

        this.nom="Volkswagen e-Golf 100 KW";

        this.anchoC = 1.799;
        this.altoC = 1.482;
        this.capacidadC = 35.8;
        this.pesoC = 1615;
        this.cdC = 0.28;
        document.getElementById("boton_vehC").innerHTML =
this.nom;

    }

    if(this.testRadioResult == "12"){

        this.nom="Volkswagen e-Golf 85 KW";

        this.anchoC = 1.799;
        this.altoC = 1.453;
        this.capacidadC = 24.2;
        this.pesoC = 1585;
        this.cdC = 0.28;
        document.getElementById("boton_vehC").innerHTML =
this.nom;

    }

    if(this.testRadioResult == "13"){

        this.nom= "Otro";
        document.getElementById("boton_vehC").innerHTML=
this.nom;

    }
}
```

```

    }

    });

    alert.present().then(() => {
        this.testRadioOpen = true;
    });

}

Iralmapa(){

    this.navCtrl.push(this.mapaPage,{
        anchoC1: this.anchoC,
        altoC1: this.altoC,
        capacidadC1 : this.capacidadC,
        personasC1 : this.personasC,
        pesoC1 : this.pesoC,
        intervaloC1 : this.intervaloC,
        cdC1 : this.cdC,
        nombre : this.nom
    });
    console.log("AnchoC:",this.anchoC); //Muestra por
pantalla el valor que se indique y el texto que se encuentre
entre comillas.

}

}

```

- Calibracion.css

```

page-calibracion {

    ion-content{

        #parrafo2{
            font-family: Cambria, Cochin, Georgia, Times,
'Times New Roman', serif;
            font-size: 20px;
            font-weight: 300;
        }

        #parrafo3{
            font-family: Cambria, Cochin, Georgia, Times,
'Times New Roman', serif;

```

```

        font-size: 20px;
        font-weight: 300;
    }

    #parrafo4{
        font-family: Cambria, Cochin, Georgia, Times,
'Times New Roman', serif;
        font-size: 20px;
        font-weight: 700;
    }

    .calibracion{
        font-weight: 700;
    }
}
}

```

- Cerrarsesion.html

```

<ion-header>

  <ion-navbar>
    <ion-title>Cerrar sesión</ion-title>
  </ion-navbar>

</ion-header>

<ion-content padding>

</ion-content>

```

- Cerrarsesion.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { TabsPage } from '../tabs/tabs';

@IonicPage()
@Component({
  selector: 'page-cerrar-sesion',
  templateUrl: 'cerrar-sesion.html',
})
export class CerrarSesionPage {

```



```

    constructor(public navCtrl: NavController, public
navParams: NavParams) {
    }

    ionViewDidLoad() {
        console.log('ionViewDidLoad CerrarSesionPage');
    }

    cerrarsesion(){
        this.navCtrl.push(TabsPage);
    }
}

```

- Inforvehiculo.html

```

<ion-header>

    <ion-navbar>
        <ion-title>Funcionamiento normal</ion-title>
    </ion-navbar>

</ion-header>

<ion-content padding>
    <p id="parrafo8"> Este es el modo de funcionamiento normal.
A él se accede si ya ha realizado
    la calibración con el vehículo que va a circular en este
modo. Si no es así, por favor,
    utilice primero el modo de calibración para poder hacer
un ajuste lo más real posible.
    </p>
    <p id="parrafo9"> Si selecciona uno de los modelos que se
indican en la lista que se muestra dándole en el siguiente
botón, se le adjudicarán algunos valores que vienen
    dados por su ficha técnica. Si ud. cree que debe
cambiarlos, podrá hacerlo. Si su vehículo no se encuentra en
la lista, seleccione "Otro".
    </p>

    <button ion-button color="danger" outline id= "boton_veh"
(click)="doRadio()">Modelo del vehículo</button>

```

```

<ion-list>

    <ion-item>
        <ion-label fixed>Crr (de calib.)</ion-label>
        <ion-input id= "crr" [(ngModel)]="crr"
type="number"></ion-input>
    </ion-item>

    <ion-item>
        <ion-label fixed>Cx (Cd)</ion-label>
        <ion-input id= "cd3" [(ngModel)]="cd3"
type="number"></ion-input>
    </ion-item>

    <ion-item>
        <ion-label>Número de pasajeros</ion-label>
        <ion-select [(ngModel)]="personasF"
id="num_pasajerosF" submitText="Ok" cancelText="Cancelar"
interface="popover">
            <ion-option value="1">1</ion-option>
            <ion-option value="2">2</ion-option>
            <ion-option value="3">3</ion-option>
            <ion-option value="4">4</ion-option>
            <ion-option value="5">5</ion-option>
        </ion-select>
    </ion-item>

    <ion-item>
        <ion-label >Velocidad máxima media de las
carreteras por las que circulará</ion-label>
        <ion-select [(ngModel)]="veloc" id="veloc"
submitText="Ok" cancelText="Cancelar" interface="popover">
            <ion-option value="50">50</ion-option>
            <ion-option value="90">90</ion-option>
            <ion-option value="100">100</ion-option>
            <ion-option value="120">120</ion-option>
        </ion-select>
    </ion-item>

</ion-list>

<button id="boton_ok" ion-button round
color="secondary" (click)="Iralmapa()">ACEPTAR</button>

```

```
</ion-content>
```

- Inforvehiculo.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { AlertController } from 'ionic-angular';
import { MapaPage } from '../mapa/mapa';

@IonicPage()
@Component({
  selector: 'page-infor-vehiculo',
  templateUrl: 'infor-vehiculo.html',
})
export class InforVehiculoPage {
  testRadioOpen: boolean;
  testRadioResult;

  mapaPage : any;

  cd3 : any;
  personas3 : any;
  crr3 : any;
  ancho3 : any;
  alto3 : any;
  peso3 : any;
  capacidad3 : any;
  intervalo3 : any;
  nombre3 : any;
  distancia3 : any;
  veloc : any;

  constructor(public navCtrl: NavController, public
navParams: NavParams, public alerCtrl: AlertController) {
    this.mapaPage=MapaPage;

    this.ancho3=navParams.data.ancho3;
    this.alto3=navParams.data.alto3;
    this.personas3=navParams.data.personas3;
    this.peso3=navParams.data.peso3;
    this.capacidad3=navParams.data.capacidad3;
    this.intervalo3=navParams.data.intervalo3;
    this.cd3=navParams.data.cd3;
```

```
this.nombre3=navParams.data.nombre3;
this.distancia3=navParams.data.distancia3;

}
Iralmapa(){

  this.navCtrl.push(this.mapaPage,{
    personas4 : this.personas3,
    crr4 : this.crr3,
    ancho4: this.ancho3,
    alto4: this.alto3,
    capacidad4 : this.capacidad3,
    peso4 : this.peso3,
    intervalo4 : this.intervalo3,
    cd4 : this.cd3,
    distancia4 : this.distancia3,
    veloc4 : this.veloc

  });
  console.log(this.cd3);
}

stpSelect() {
  console.log('STP selected');
}

ionViewDidLoad() {
  console.log('ionViewDidLoad InforVehiculoPage');
  document.getElementById("boton_veh").innerHTML =
this.nombre3;
}

doRadio() {

  let alert = this.alerCtrl.create();
  alert.setTitle('Vehículo');

  alert.addInput({
    type: 'radio',
    label: 'Kia SOUL EV',
    value: '1',
    checked: true
  });

  alert.addInput({
    type: 'radio',
    label: 'Peugeot iON',
```

```
        value: '2'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'Renault ZE',  
        value: '3'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'Hyundai IONIC HE',  
        value: '4'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'Renault ZOE',  
        value: '5'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'Nissan LEAF Acenta/Tekna/Visia/Visia+',  
        value: '6'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'BMW i3',  
        value: '7'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'Smart ForTwo eD Cabrio/Coupé',  
        value: '8'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'Tesla S 75D',  
        value: '9'  
    });  
  
    alert.addInput({  
        type: 'radio',  
        label: 'Tesla S P100D',  
        value: '10'  
    });
```

```
});

alert.addInput({
  type: 'radio',
  label: 'Volkswagen e-Golf 100 KW',
  value: '11'
});

alert.addInput({
  type: 'radio',
  label: 'Volkswagen e-Golf 85 KW',
  value: '12'
});

alert.addInput({
  type: 'radio',
  label: 'Otro',
  value: '13'
});

alert.addButton('Cancelar');
alert.addButton({
  text: 'Ok',
  handler: data => {
    console.log('Radio data:', data);
    this.testRadioOpen = false;
    this.testRadioResult = data; //data es el valor, es
    decir, el número del vehículo elegido

    if(this.testRadioResult == "1"){

      var nom="Kia SOUL EV";
      this.cd3 = 0.33;
      document.getElementById("boton_veh").innerHTML= nom;

    }

    if(this.testRadioResult == "2"){

      var nom2="Peugeot iON";
      this.cd3 = 0.33;
      document.getElementById("boton_veh").innerHTML = nom2;

    }

    if(this.testRadioResult == "3"){

      var nom3="Renault ZE";
```

```
this.cd3 = 0.33;
document.getElementById("boton_veh").innerHTML = nom3;

}

if(this.testRadioResult == "4"){

    var nom4="Hyundai IONIQ BEV";
    this.cd3 = 0.24;
    document.getElementById("boton_veh").innerHTML = nom4;

}

if(this.testRadioResult == "5"){

    var nom5="Renault ZOE";

    this.cd3 = 0.34;
    document.getElementById("boton_veh").innerHTML = nom5;

}

if(this.testRadioResult == "6"){

    var nom6="Nissan LEAF Acenta/Tekna/Visia/Visia+";
    this.cd3 = 0.29;
    document.getElementById("boton_veh").innerHTML = nom6;

}

if(this.testRadioResult == "7"){

    var nom7="BMW i3";
    this.cd3 = 0.29;
    document.getElementById("boton_veh").innerHTML = nom7;

}

if(this.testRadioResult == "8"){

    var nom8="Smart ForTwo eD Cabrio/Coupé";

    this.cd3 = 0.29;
    document.getElementById("boton_veh").innerHTML = nom8;
```

```
}

if(this.testRadioResult == "9"){

    var nom9="Tesla S 75D";

    this.cd3 = 0.24;
    document.getElementById("boton_veh").innerHTML = nom9;

}

if(this.testRadioResult == "10"){

    var nom10="Tesla S P100D";

    this.cd3 = 0.24;
    document.getElementById("boton_veh").innerHTML = nom10;

}

if(this.testRadioResult == "11"){

    var nom11="Volkswagen e-Golf 100 KW";

    this.cd3 = 0.28;
    document.getElementById("boton_veh").innerHTML = nom11;

}

if(this.testRadioResult == "12"){

    var nom12="Volkswagen e-Golf 85 KW";

    this.cd3 = 0.28;
    document.getElementById("boton_veh").innerHTML = nom12;

}

}

});
```



```

    alert.present().then(() => {
        this.testRadioOpen = true;
    });

    if(this.testRadioResult == "13"){

        var nom13= "Otro";
        document.getElementById("boton_veh").innerHTML= nom13;

    }

    }

}

```

- Inforvehiculo.css

```

page-infor-vehiculo {

    ion-content {

        #parrafo8 {
            font-family: Cambria, Cochin, Georgia, Times,
'Times New Roman', serif;
            font-size: 20px;
            font-weight: 300;
        }

        #parrafo9 {
            font-family: Cambria, Cochin, Georgia, Times,
'Times New Roman', serif;
            font-size: 20px;
            font-weight: 300;
        }

    }

}

```

- Mapa.html

```
<ion-header>
```

```

<ion-navbar>
  <ion-buttons end>
    <button ion-button icon-only
(click)="iramenumapa()">
      <ion-icon name="ios-list-box-outline"></ion-icon>
    </button>
  </ion-buttons>
  <ion-title>Mapa</ion-title>
  <ion-buttons start>
    <button ion-button icon-only>
      <ion-icon name="search"></ion-icon>
    </button>
  </ion-buttons>

</ion-navbar>
<ion-item id="peajes">
  <ion-label>Evitar peajes si es posible</ion-label>
  <ion-toggle color = "energized" checked="true"></ion-
toggle> //Es un botón de distinto tipo que el de button.
</ion-item>

<br>

</ion-header>

<ion-content padding>

  <input id="pac-input" class="controls" type="text"
placeholder="Introduce una localización">
  //Placeholder para que el texto se introduzca en el interior
del input y al darle para escribir desaparezca.
  <div id="map"></div>
  <div id="infowindow-content">
    <span id="place-name" class="title"></span><br>
    Place ID: <span id="place-id"></span><br>
    <span id="place-address"></span>
  </div>
  <!-->
  <br>
  <button id="trazarRuta" ion-button round block
color="primary" (click)="calculateRoute()">TRAZAR
RUTA</button>
  <br>
  <button id="boton_calibrar" ion-button round
color="danger" (click)="Calculo2()">CALIBRAR</button>

```

```

    &nbsp;

    <button id="boton_func" ion-button round
color="secondary" (click)="Calculo()">FUNCIONAMIENTO <br>
NORMAL</button>

    <ion-scroll scrollY="true" class="scroll">
        <div padding-horizontal id="panel"></div>
    </ion-scroll>

    <ion-item>
        <ion-label>Distancia a recorrer (km)</ion-label>
        <ion-input type="number" [(ngModel)]="distan"
id="distan" disabled></ion-input>
    </ion-item>

</ion-content>

```

- Mapa.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { MenuPage } from '../menu/menu';
import { CalculoEficienciaPage } from '../calculo-
eficiencia/calculo-eficiencia';
import { CalculoCortoPage } from '../calculo-corto/calculo-
corto';
import { Geolocation, Geoposition } from '@ionic-
native/geolocation';

declare var google; //Variable global.

@IonicPage()
@Component({
    selector: 'page-mapa',
    templateUrl: 'mapa.html',
})
export class MapaPage {

    map : any;
    s : any;
    p : any;
    prueba : number;

```

```
calculoeficienciaPage : any;
calculoCortoPage : any;

    ancho4 : any;
    alto4 : any;
    personas4 : any;
    peso4 : any;
    capacidad4 : any;
    intervalo4 : any;
    crr4 : any;
    cd4 : any;
    distancia4 : any;
    veloc4 : any;

//Calibración
    anchoC1 : any;
    altoC1 : any;
    personasC1 : any;
    pesoC1 : any;
    capacidadC1 : any;
    intervaloC1 : any;
    cdC1 : any;
    nombre : any;
    altura : any;

//Crear ruta
    directionsService: any = null;
    directionsDisplay: any = null;
    bounds: any = null;
    elevator : any;
    myLatLng: any;
    latitLong: any;
    waypoints: any[];
    long: any;
    latit: any;
    place : any;

    distan : any;

//

    constructor(public navCtrl: NavController, public
navParams: NavParams, public geolocation: Geolocation) {
        this.s= document.getElementById("hy");
        this.p = document.getElementById("peajes");
        this.prueba = 0;

        this.calculoeficienciaPage=CalculoEficienciaPage;
```

```
this.calculoCortoPage=CalculoCortoPage;

//Calibración
this.anchoC1=navParams.data.anchoC1;
this.altoC1=navParams.data.altoC1;
this.personasC1=navParams.data.personasC1;
this.pesoC1=navParams.data.pesoC1;
this.capacidadC1=navParams.data.capacidadC1;
this.intervaloC1=navParams.data.intervaloC1;
this.cdC1=navParams.data.cdC1;
this.nombre=navParams.data.nombre;

//Func. normal
this.personas4=navParams.data.personas4;
this.cd4=navParams.data.cd4;
this.crr4=navParams.data.crr4;
this.ancho4=navParams.data.ancho4;
this.alto4=navParams.data.alto4;
this.peso4=navParams.data.peso4;
this.capacidad4=navParams.data.capacidad4;
this.intervalo4=navParams.data.intervalo4;
this.distancia4=navParams.data.distancia4;
this.veloc4=navParams.data.veloc4;

//
this.directionsService = new
google.maps.DirectionsService();
this.directionsDisplay = new
google.maps.DirectionsRenderer();
this.bounds = new google.maps.LatLngBounds(); //Variable
que sirve para definir los límites.
this.elevator = new google.maps.ElevationService;
//Variable para la altura.
}

Calculo(){
this.navCtrl.push(this.calculoeficienciaPage,{
  ancho5: this.ancho4,
  alto5: this.alto4,
  capacidad5 : this.capacidad4,
  personas5 : this.personas4,
  peso5 : this.peso4,
  intervalo5 : this.intervalo4,
  crr5 : this.crr4,
  cd5 : this.cd4,
  distancia5 : this.distancia4,
  altura5 : this.altura,
```

```

        veloc5 : this.veloc4
    });
}
//Calibración
Calculo2(){
    this.navCtrl.push(this.calculoCortoPage,{
        anchoC2: this.anchoC1,
        altoC2: this.altoC1,
        capacidadC2 : this.capacidadC1,
        personasC2 : this.personasC1,
        pesoC2 : this.pesoC1,
        intervaloC2 : this.intervaloC1,
        cdC2 : this.cdC1,
        nombre2 : this.nombre,
        altura2 : this.altura,
        distancia2 : this.distan
    });
}

iramenumapa(){
    this.navCtrl.push(MenuPage);
}
ionViewDidLoad() {
    this.getPosition();
    console.log('ionViewDidLoad MapaPage');
}

getPosition(): any { //Da las coordenadas del dispositivo
en latitud y longitud
    this.geolocation.getCurrentPosition().then(response => {
        this.loadMap(response); //Si consigue conocer la
localización, llama al método loadMap.
    })
    .catch(error =>{ //Si no lo consigue, lanza un mensaje de
error por pantalla.
        console.log(error);
    })
}

loadMap(position: Geoposition){ //Posiciona al mapa en las
coordenadas de longitud y latitud en las que está el
dispositivo
    let latitude = position.coords.latitude;
    let longitude = position.coords.longitude;
    this.altura = position.coords.altitude;
    console.log(latitude, longitude);
    console.log("Altitud", this.altura);
}

```

```
// Creamos el mapa a partir de un HTMLElement.
let mapEle: HTMLElement = document.getElementById('map');
//Creamos el panel de instrucciones a partir de un
HTMLElement.
let panelEle: HTMLElement =
document.getElementById('panel');//

// Creamos un objeto LatLng que contiene la latitud y
longitud
this.myLatLng = {lat: latitude, lng: longitude};
console.log("MyLatLng es:",this.myLatLng);

// Creamos el mapa con sus propiedades
this.map = new google.maps.Map(mapEle, {
  center: this.myLatLng,
  zoom: 12
});

this.directionsDisplay.setMap(this.map);
this.directionsDisplay.setPanel(panelEle);
//Una vez se produzca un evento, se pinta el marcador de
la posición actual en el mapa.
google.maps.event.addListenerOnce(this.map, 'idle', () =>
{
  let marker = new google.maps.Marker({
    position: this.myLatLng,
    map: this.map,
    title: 'Estás aquí',
    draggable : true,
  }); //Graggable para poder arrastrarlo sobre el mapa.

  mapEle.classList.add('show-map');

  var input = document.getElementById('pac-input');

//Cuadro de texto para buscar

  var autocomplete = new
google.maps.places.Autocomplete(input); //Para que
autocomplete la búsqueda
  autocomplete.bindTo('bounds', this.map);

(this.map).controls[google.maps.ControlPosition.TOP_LEFT].pus
h(input); //El cuadro irá arriba a la izquierda
```

```
var infowindow = new google.maps.InfoWindow(); //Cuadro
de información sobre el marker nuevo

var marker1 = new google.maps.Marker({ //Marker nuevo
  map: this.map,
  draggable: true,
  animation: 'DROP',
  title: "Destino obtenido del cuadro de búsqueda"
});

marker1.addListener('click', () => { //Al hacer click
sobre él se abre el cuadro de información
  infowindow.open(this.map, marker1);
});

console.log("hola1");
console.log("prueba0:", this.prueba)
//Una vez se le da a la sugerencia correcta, se ejecuta la
siguiente función.
autocomplete.addListener('place_changed', () => {
  console.log("Entrar en place_changed");
  infowindow.close();
  this.place = autocomplete.getPlace(); //Va al lugar que
se ha detectado con el autocompletar, getPlace devuelve el id
del sitio

  if (!this.place.geometry) {
    console.log("Salimos porque no tenemos geometría");
    return;
  }
//Latitud y longitud del nuevo marcador del destino.
this.latit = this.place.geometry.location.lat();
console.log("Latit vale:", this.latit);
this.long = this.place.geometry.location.lng();

  this.latitLong = {lat: this.latit, lng:
this.long}; //Devuelve un objeto

  console.log("latitLong vale:", this.latitLong);

  if (this.place.geometry.viewport) {
    console.log("Entra en viewport y es:",
this.place.geometry.viewport);

  } else {
    this.map.setCenter(this.place.geometry.location);
    this.map.setZoom(17);
  }
}
```



```
    }

    marker1.setPosition(this.place.geometry.location);
    marker1.setVisible(true);

    document.getElementById('place-name').textContent =
this.place.name;
    document.getElementById('place-id').textContent =
this.place.place_id;
    document.getElementById('place-address').textContent
=
    this.place.formatted_address;

infowindow.setContent(document.getElementById('infowindow-
content'));
    infowindow.open(this.map, marker1);

    this.prueba = 1;
    console.log("Sale del autocomplete (prueba1):",
this.prueba);

    console.log("Long vale:", this.long);

}); //final función de autocomplete

}); //Final de addlistener primero, el del marker

} //Final de loadMap

//Mediante la siguiente función, se traza la ruta a seguir.

calculateRoute(){
    console.log("Entramos en calcular ruta");
    console.log("Pasa bounds");
    this.map.fitBounds(this.bounds);
    console.log("Pasa fitbounds");

    this.distan = document.getElementById("latitlong");

    this.directionsService.route({
        origin: new google.maps.LatLng(this.myLatLng.lat,
this.myLatLng.lng),
```

```

        destination: new google.maps.LatLng (this.latit,
this.long),
        travelMode: google.maps.TravelMode.DRIVING,
        avoidTolls: true
    }, (response, status)=> {
        if (status === google.maps.DirectionsStatus.OK) {
            console.log(response);

            this.directionsDisplay.setDirections(response);
            this.distan =
response.routes[0].legs[0].distance.value / 1000; //Distancia
entre los dos puntos
        } else {
            alert('Could not display directions due to: ' +
status);
        }
    });
}
}

```

- Mapa.css

```

page-mapa {
    ion-content{ //Se fijan las características del mapa,
entre ellas el alto y el ancho, imprescindible para que el
mapa se vea en la aplicación.
        background: rgb(229, 227, 223);
        #map {
            width: 100%;
            height: 100%;
            opacity: 0;
            transition: opacity 150ms ease-in;
            display: block;
            &.show-map{
                opacity: 1;
            }
        }
        .scroll{
            height: calc(100% - 250px);
        }
        html, body {
            height: 100%;
            margin: 0;
        }
    }
}

```

```
padding: 0;
}
.controls {
background-color: #fff;
border-radius: 2px;
border: 1px solid transparent;
box-shadow: 0 2px 6px rgba(0, 0, 0, 0.3);
box-sizing: border-box;
font-family: Roboto;
font-size: 15px;
font-weight: 300;
height: 29px;
margin-left: 17px;
margin-top: 10px;
outline: none;
padding: 0 11px 0 13px;
text-overflow: ellipsis;
width: 400px;
}

.controls:focus {
border-color: #4d90fe;
}
.title {
font-weight: bold;
}
#infowindow-content {
display: none;
}
#map #infowindow-content {
display: inline;
}
}
```

- Menu.html

```
<ion-header>
<ion-navbar>
  <ion-title>Menú</ion-title>
</ion-navbar>
</ion-header>
```

```

<ion-content padding>

  <ion-list>

    <button ion-button medium clear color="dark" (click)=
"iniciarsesion()" icon-start>
      <ion-icon name= "ios-log-in"></ion-icon>
      Iniciar sesión
    </button>
    <br>
    <hr>
    <button ion-button color="dark" (click)=
"registrarse()" medium clear icon-start>
      <ion-icon name= "md-person-add"></ion-icon>
      Registrarse
    </button>
    <br>
    <hr>
    <button ion-button medium color="dark" clear (click)=
"nuevaruta()">
      <ion-icon name= "ios-create-outline"></ion-icon>
      &nbsp;
      Comenzar nueva ruta
    </button>
    <br>
    <hr>
    <button ion-button medium color="dark" clear (click)=
"cerrarapp()"> <!--Icono a la izq: primero icono y después
texto-->
      <ion-icon name= "ios-log-out"></ion-icon>
      &nbsp; <!--Espacio-->
      Cerrar
    </button>
    <br>
    <hr>

  </ion-list>

```

- Menu.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { SobremenuPage } from '../sobremenu/sobremenu';
import { HomePage } from '../home/home';
import { InicioSesionPage } from '../inicio-sesion/inicio-
sesion';

```

```
import { RegistroPage } from '../registro/registro';
import { TabsPage } from '../tabs/tabs';
import { MapaPage } from '../mapa/mapa';

@IonicPage()

@Component({
  selector: 'page-menu',
  templateUrl: 'menu.html',
})
export class MenuPage {

  usuario : any;
  iniciosesionPage : any;

  constructor(public navCtrl: NavController, public
navParams: NavParams) {

this.usuario = navParams.data.usuario;
this.iniciosesionPage = InicioSesionPage

  }

//Las variables no son comunes a todas las páginas, hay que
importarlas.

  irahome(){
    this.navCtrl.push(HomePage);
  }

  iniciarsesion(){
    this.navCtrl.push(this.iniciosesionPage,{
      usuario1 : this.usuario
    });
  }

  registrarse(){
    this.navCtrl.push(RegistroPage);
  }

  nuevaruta(){
    this.navCtrl.push(MapaPage);
  }

  cerrarapp(){
    this.navCtrl.push(TabsPage); //Vuelve a la página de
inicio
  }
}
```

```
ionViewDidLoad() {  
  console.log('ionViewDidLoad MenuPage');  
}  
  
}
```

- Iniciosesion.html

```
<ion-header>  
  
  <ion-navbar>  
    <ion-title>Iniciar sesión</ion-title>  
  </ion-navbar>  
  
</ion-header>  
  
<ion-content padding>  
  
  <ion-list>  
    <ion-item>  
      <ion-label stacked>Correo electrónico</ion-label>  
      <ion-input type="text" id="id_email" ></ion-input>  
    </ion-item>  
  
    <ion-item>  
      <ion-label stacked>Contraseña</ion-label>  
      <ion-input type="password" name="password" required minlength="6"></ion-input>  
    </ion-item>  
  
    <div padding-left padding-right padding-top>  
      <button ion-button block (click)="ponernombre()">Acceder</button>  
    </div>  
  </ion-list>  
  
</ion-content>
```

- Iniciosesion.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';
import { TabsPage } from '../tabs/tabs';

@IonicPage()
@Component({
  selector: 'page-inicio-sesion',
  templateUrl: 'inicio-sesion.html',
})
export class InicioSesionPage {

  usuario1 : any;
  constructor(public navCtrl: NavController, public
navParams: NavParams) {

    this.usuario1=navParams.data.usuario1;
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad InicioSesionPage');
  }

  ponernombre(){

    let a = document.getElementById("id_email").innerHTML;
    this.usuario1 = a;
    this.navCtrl.push(TabsPage);
    console.log(a);
  }
}

```

- o Registro.html

```

<ion-header>

  <ion-navbar>
    <ion-title>Nuevo usuario</ion-title>
  </ion-navbar>

</ion-header>

<ion-content padding>

```

```

<ion-list>
  <ion-item>
    <ion-label stacked>Correo electrónico</ion-label>
    <ion-input type="email" autocorrect="off"
autocapitalize="none" name="email" pattern="[a-z0-9._%+-
]+@[a-z0-9.-]+\.[a-z]{2,3}$"
      required></ion-input>
    </ion-item>

    <ion-item>
      <ion-label stacked>Contraseña</ion-label>
      <ion-input type="password" name="password"
required minlength="6"></ion-input>
    </ion-item>

    <div padding-left padding-right padding-top>

      <button ion-button block>Registrarse</button>
    </div>
  </ion-list>
</ion-content>

```

- Registro.ts

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-
angular';

@IonicPage()
@Component({
  selector: 'page-registro',
  templateUrl: 'registro.html',
})
export class RegistroPage {

  constructor(public navCtrl: NavController, public
navParams: NavParams) {
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad RegistroPage');
  }
}

```


Nota: los archivos “.css” de cada página que no se incorporan, es porque no se han utilizado.

15. Bibliografía

A continuación se muestra la bibliografía usada en este proyecto que no ha sido citada:

1. Regulador eléctrico. Página web oficial de Renault.
2. Motores eléctricos.
 - Fraile, J., *Máquinas eléctricas*, España, Mc Graw Hill.
 - Gómez, J.F., *Máquinas de corriente continua*, Depto. de Física básica, Universidad de La Laguna.
 - Rippel W. “Induction Versus DC Brushless Motors.” *Tesla*, 9 enero 2007.
[https://www.tesla.com/es_ES/blog/induction-versus-dc-brushless-motors?redirect=no]
3. Tribunal de cuentas europeo, “Acción de la UE en materia de energía y cambio climático” [<http://publications.europa.eu/webpub/eca/lr-energy-and-climate/es/>].
4. Funcionamiento de las baterías eléctricas.
 - Lavela, P., Tirado, J.L., 1999, *Baterías avanzadas*, España, Universidad de Córdoba.
5. Funcionamiento del freno regenerativo.
 - Domínguez, E.J., Ferrer J., 2012, *Sistemas de transmisión y frenado*, España, Editec.

6. Funcionamiento de un vehículo eléctrico.

Vázquez, Arturo. “Así funciona el motor de un coche eléctrico.” *ABC* [Madrid] 25 septiembre 2017. [https://www.abc.es/motor/reportajes/abci-funciona-motor-coche-electrico-201702011409_noticia.html]

7. Coeficiente de resistencia a la aerodinámica.

- Avallone A. Eugene Marks, 1995, *Marks manual del ingeniero mecánico*, Mc Graw Hill.

- Da Riva I., *Aerodinámica*, Laboratorio de Aerodinámica, Universidad Politécnica de Madrid.

8. Rodadura y deslizamiento.

Teresa Martín Blas y Ana Serrano Fernández, España, Universidad Politécnica de Madrid. [<http://www2.montes.upm.es/dptos/digfa/cfisica/solido/rodadura.html>]

Referencias bibliográficas citadas a lo largo del texto:

[1] Agencia Internacional de la Energía (IEA): Global EV Outlook 2016, “*Beyond one million electric cars*”.

[2] Plataforma colaborativa *Electromaps*.

[3] Valøen, Lars Ole and Shoesmith, Mark I. (2007). “*The effect of PHEV and HEV duty cycles on battery and battery pack performance*”.

[4] AAA: American Automobile Association, “*Extreme temperatures affect electric vehicle*” [<https://newsroom.aaa.com/2014/03/extreme-temperatures-affect-electric-vehicle-driving-range-aaa-says/>].

[5] SciELO¹⁶, “*Methodology for fast calculation of agricultural tractors tires' dimensions*”, Universidad Politécnica de Madrid.

[6] Turiel A., “*The Oil Crush*”.

¹⁶ SciELO: Biblioteca virtual formada por una colección de revistas científicas españolas de ciencias de la salud seleccionadas de acuerdo a unos criterios de calidad preestablecidos.

[7] -IDAE, Estrategia de Movilidad Española y Asociación Europea para Vehículos Eléctricos a Batería.

-Informe Básico 2008 sobre Sectores de la Energía de la CNE.

-Página oficial de Renault.