



TRABAJO FIN DE GRADO
INGENIERÍA EN INFORMÁTICA

Plataforma de economía colaborativa para la movilidad entre España y Senegal

Autor

Iñaki Fernández Janssens de Varebeke

Tutora

Rosana Montes Soldado



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

Granada, Septiembre de 2015



Plataforma de economía colaborativa para la movilidad entre España y Senegal

Autor

Iñaki Fernández Janssens de Varebeke

Tutora

Rosana Montes Soldado

Plataforma de economía colaborativa para la movilidad entre España y Senegal

Iñaki Fernández Janssens de Varebeke

Palabras clave: carsharing, paquetería, economía colaborativa, migración, Senegal

Resumen

Este proyecto contempla la creación de una aplicación web para favorecer la movilidad de los flujos migratorios internacionales, más concretamente entre España y Senegal, apoyándose en la *economía colaborativa* ^[T1].

Esta aplicación permitirá aprovechar los desplazamientos que realiza en coche un usuario entre España y Senegal para permitir a otros usuarios participar en el viaje como pasajeros compartiendo gastos de transporte, por lo que todos los usuarios se benefician con un ahorro de costes del viaje.

Además de lo anteriormente dicho, la aplicación permitirá el transporte de paquetería pequeña, de modo que cualquier usuario podrá solicitar que un conductor cargue como equipaje su paquete para transportarlo de forma mucho más económica entre países.

Estos viajes serán regulados y puntuados por la propia comunidad de la aplicación, de modo que se creará una valoración de cada usuario que realice viajes según la satisfacción de los pasajeros que ha llevado, creándose un sistema que medirá socialmente la confianza sobre cada usuario.

Se creará por tanto un sistema integrado por una base de conocimiento, un modelo de descripción de rutas y un sistema cualitativo de valoración de los distintos elementos implicados en el servicio.

La pagina web se desarrollará haciendo uso de la innovadora biblioteca "*Polymer*" ^[5.1.2.2] (la cual introduce los denominados "*web components*" ^[T2] como un concepto revolucionario a la forma de desarrollar sitios web hasta el momento) y será una aplicación completamente del lado del cliente, haciendo un uso intensivo de la tecnología asíncrona *AJAX* ^[A2] y del soporte *Javascript* ^[5.1.1.1] de los navegadores actuales. Esto permitirá disponer de una aplicación moderna, cómoda y versátil, mejorando la experiencia de usuario a la vez que permitiendo realizar una gran optimización en el uso de recursos.

Esta aplicación solicitará los datos a un servidor *REST* ^[A3], el cual, a su vez, hará uso de una base de datos *MySQL* ^[5.1.1.2] para el almacenamiento de los mismos, independizando completamente la aplicación del servidor de consulta y este del modo de almacenamiento.

Iñaki Fernández Janssens de Varebeke

Keywords: carsharing, package, collaborative economy, migration, Senegal

Abstract

This project concerns the creation of a web application to make easier the international migration flows mobility, more concretely, between Spain and Senegal, relying on the collaborative economy.

This application will give everyone the opportunity to take advantage of the travels a user does between Spain and Senegal, allowing other users be part of the trip as passenger sharing the transport costs. With this system, both users, drivers and passengers, will benefit from the travel saving trips costs.

In addition to the previous statement, the application will allow to use the already planned trips as a small package transport service in the way that every user will be able to request a driver to take a package as luggage a small item to be carried between countries without the need of a external package service, being more "economy friendly" and so, saving money.

The trips are being regulated and assessed by the own application community so that every user that takes part of a trip can assess it. This assessments are being converted in a per driver score, creating a system that will measure the confidence about each driver.

There will be created, thus, a knowledge base integrated system, a route description model and a qualitative assessment system of the different elements involved with the service.

The web site will be developed with the use of the innovative library "Polymer" (which introduces the called "web components" as a revolutionary concept to the way the web sites are developed till the moment) and will be a full client side application, making an intensive use of the *AJAX* asynchronous technology and the current browsers *Javascript* support. This will allow to have a modern, comfortable and versatile application, improving the user's experience while allowing to perform a great optimization on terms of IT resources.

This application will request the data to a *REST* server, which, in turn, will make use of a *MySQL* database for it storage, separating completely the client application from the query server and this from the storage mode.

Yo, **Iñaki Fernández Janssens de Varebeke**, alumno de la titulación *Grado en Ingeniería Informática* de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 78552790Q, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: *Iñaki Fernández Janssens de Varebeke*

Granada a 11 de Septiembre de 2015.

Dña. **Rosana Montes Soldado**, profesora del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado *Plataforma de economía colaborativa para la movilidad entre España y Senegal*, ha sido realizado bajo su supervisión por **Iñaki Fernández Janssens de Varebeke**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 11 de Septiembre de 2015.

La tutora:

Rosana Montes Soldado

Agradecimientos

Me gustaría utilizar este espacio para agradecer principalmente a mi pareja, Ely, por estar siempre a mi lado, ayudarme y darme ánimos en todo lo posible.

Índice

RESUMEN.....	5
ABSTRACT	6
1. INTRODUCCIÓN.....	20
1.1 Motivación	21
1.2 Estado del arte	23
1.2.1 <i>BlaBlaCar</i> ^{[B1][B2]}	24
1.2.2 <i>Amovens</i> ^{[B3][B4]}	28
1.2.3 <i>RoadSharing</i> ^[B5]	30
1.2.4 <i>Conclusiones</i>	31
1.3 Estructura del documento	32
1.4 Objetivos del proyecto	34
2. PLANIFICACIÓN.....	37
2.1 Fases.....	38
2.1.1 <i>Fase inicial</i>	38
2.1.2 <i>Fase de análisis</i>	38
2.1.3 <i>Fase de desarrollo</i>	39
2.1.4 <i>Fase de prueba y corrección</i>	41
2.1.5 <i>Redacción de la memoria</i>	41
2.2 Presupuesto	43
2.2.1 <i>Recursos</i>	43
2.2.2 <i>Costes</i>	44
3. ANÁLISIS.....	47
3.1 Especificación de requisitos	48
3.1.1 <i>Requisitos funcionales</i>	48
3.1.2 <i>Restricciones semánticas</i>	57
3.1.3 <i>Requisitos de rendimiento</i>	58
3.2 Modelos de caso de uso	59
3.2.1 <i>Diagrama de casos de uso</i>	59
3.3 Modelos de comportamiento	61
4. DISEÑO.....	73
4.1 Diseño de Base de Datos.....	74
4.1.1 <i>Diagrama Entidad-Relación inicial</i>	74
4.1.2 <i>Diagrama de Entidad-Relación Mejorado</i>	76
4.1.3 <i>Tablas</i>	78

4.2	Diseño de interfaz	93
4.2.1	Área principal.....	93
4.2.2	Zona de perfil / registro	97
4.2.3	Viajes	101
4.2.4	Paquetes.....	106
4.2.5	Módulos	108
4.2.6	Responsibilidad ^[B12]	111
5.	IMPLEMENTACIÓN.....	112
5.1	Herramientas utilizadas.....	113
5.1.1	Lenguajes utilizados	113
5.1.2	Tecnologías usadas.....	115
5.1.3	Servidor.....	116
5.1.4	Programas externos.....	117
5.2	Fases del desarrollo	118
5.2.1	Desarrollo de la base de datos ^[B17]	118
5.2.2	Creación de la API REST ^[B8]	119
5.2.3	Estructuración de la aplicación web ^{[B17] [B9]}	127
5.2.4	Sistemas de sesiones ^{[B8] [B9]}	128
5.2.5	Sistema de notificaciones.....	129
5.2.6	Conexión con la API de Google ^{[B14] [B15] [B16]}	130
5.3	Diseño final de la interfaz	133
5.3.1	Área principal.....	133
5.3.2	Zona de perfil / registro	136
5.3.3	Viajes	143
5.3.4	Paquetes.....	150
5.3.5	Módulos	154
6.	CONCLUSIONES	158
6.1	Objetivos alcanzados.....	159
6.2	Lecciones aprendidas	162
6.3	Vías futuras	163
7.	BIBLIOGRAFÍA	164
7.1	Enlaces de interés.....	165
8.	ANEXOS.....	166
8.1	Código fuente y ejecutable.....	167
8.2	Glosario de términos	168
8.2.1	Términos ^[B17]	168
8.2.2	Acrónimos ^[B17]	169

Índice de figuras

IMÁGENES

Imagen 1 -	Mapa de rutas migrantes Europa-África	21
Imagen 2 -	Diagrama de Gannt	42
Imagen 3 -	Diagrama de caso de uso.....	60
Imagen 4 -	Diagrama de secuencia (Registro)	61
Imagen 5 -	Diagrama de secuencia (Inicio de sesión).....	61
Imagen 6 -	Diagrama de secuencia (Cerrar sesión).....	62
Imagen 7 -	Diagrama de secuencia (Editar perfil)	62
Imagen 8 -	Diagrama de secuencia (Ver viaje)	62
Imagen 9 -	Diagrama de secuencia (Buscar viaje)	63
Imagen 10 -	Diagrama de secuencia (Crear viaje)	63
Imagen 11 -	Diagrama de secuencia (Actualizar viaje)	64
Imagen 12 -	Diagrama de secuencia (Puntuar viaje).....	64
Imagen 13 -	Diagrama de secuencia (Ver lista de viajes de un usuario).....	65
Imagen 14 -	Diagrama de secuencia (Crear solicitud)	65
Imagen 15 -	Diagrama de secuencia (Borrar solicitud).....	66
Imagen 16 -	Diagrama de secuencia (Aceptar solicitud)	66
Imagen 17 -	Diagrama de secuencia (Rechazar solicitud)	66
Imagen 18 -	Diagrama de secuencia (Lista de solicitudes)	67
Imagen 19 -	Diagrama de secuencia (Ver paquete)	67
Imagen 20 -	Diagrama de secuencia (Crear paquete).....	67
Imagen 21 -	Diagrama de secuencia (Borrar paquete)	68
Imagen 22 -	Diagrama de secuencia (Actualizar paquete).....	68
Imagen 23 -	Diagrama de secuencia (Lista de paquetes)	69
Imagen 24 -	Diagrama de secuencia (Crear coche).....	69
Imagen 25 -	Diagrama de secuencia (Borrar coche)	69
Imagen 26 -	Diagrama de secuencia (Lista de coches)	70
Imagen 27 -	Diagrama de secuencia (Perfil de usuario)	70
Imagen 28 -	Diagrama de secuencia (Lista de usuarios favoritos).....	70
Imagen 29 -	Diagrama de secuencia (Añadir favorito)	71
Imagen 30 -	Diagrama de secuencia (Eliminar favorito)	71
Imagen 31 -	Diagrama de secuencia (Leer notificación)	72
Imagen 32 -	Diagrama de secuencia (Borrar notificación)	72

Imagen 33 -	Diseño BD – Diagrama Entidad-Relación.....	75
Imagen 34 -	Diseño BD – Diagrama Entidad-Relación mejorado	77
Imagen 35 -	Tabla BD - route	78
Imagen 36 -	Tabla BD - location.....	78
Imagen 37 -	Tabla BD - trip	79
Imagen 38 -	Tabla BD - user.....	80
Imagen 39 -	Tabla BD - car.....	80
Imagen 40 -	Tabla BD - package.....	81
Imagen 41 -	Tabla BD - notification	82
Imagen 42 -	Tabla BD - requests.....	83
Imagen 43 -	Tabla BD - assessment.....	84
Imagen 44 -	Tabla BD - message	85
Imagen 45 -	Tabla BD - owners	86
Imagen 46 -	Tabla BD - routePoints	87
Imagen 47 -	Tabla BD - userTrips.....	88
Imagen 48 -	Tabla BD - packageTrips	89
Imagen 49 -	Tabla BD - favorites.....	90
Imagen 50 -	Tabla BD - tripPoints	91
Imagen 51 -	Tabla BD - packageRequests	92
Imagen 52 -	Diseño interfaz – Área principal.....	93
Imagen 53 -	Diseño interfaz – Barra superior	94
Imagen 54 -	Diseño interfaz – Barra superior sin sesión iniciada.....	94
Imagen 55 -	Diseño interfaz – Barra superior con sesión iniciada.....	94
Imagen 56 -	Diseño interfaz – Botón de perfil	94
Imagen 57 -	Diseño interfaz – Botón de notificaciones	95
Imagen 58 -	Diseño interfaz –Notificaciones no leídas	95
Imagen 59 -	Diseño interfaz – Notificaciones leídas vs. no leídas	95
Imagen 60 -	Diseño interfaz- Menú lateral	96
Imagen 61 -	Diseño interfaz – Página de registro	97
Imagen 62 -	Diseño interfaz – Formulario de logueo	98
Imagen 63 -	Diseño interfaz – Perfil propio.....	98
Imagen 64 -	Diseño interfaz – Editar perfil.....	99
Imagen 65 -	Diseño interfaz - Estrellas	99
Imagen 66 -	Diseño interfaz – Viajes de otro usuario	100
Imagen 67 -	Diseño interfaz – Creación de viajes.....	101
Imagen 68 -	Diseño interfaz – Localización sin parada	101
Imagen 69 -	Diseño interfaz – Localización con parada	101

Imagen 70 -	Diseño interfaz – Lista de viajes	102
Imagen 71 -	Diseño interfaz – Información de viaje.....	103
Imagen 72 -	Diseño interfaz – Solicitud sin paquete	104
Imagen 73 -	Diseño interfaz – Solicitud con paquete	104
Imagen 74 -	Diseño interfaz – Lista de paquetes en solicitud.....	105
Imagen 75 -	Diseño interfaz – Información paquete	106
Imagen 76 -	Diseño interfaz – Creación de paquete	107
Imagen 77 -	Diseño interfaz – Módulo viaje.....	108
Imagen 78 -	Diseño interfaz – Módulo solicitud.....	108
Imagen 79 -	Diseño interfaz – Módulo coche	109
Imagen 80 -	Diseño interfaz – Módulo paquete	109
Imagen 81 -	Diseño interfaz – Módulo usuario favorito	110
Imagen 82 -	Diseño interfaz – Módulo comentario	110
Imagen 83 -	Interfaz – Área principal	133
Imagen 84 -	Interfaz – Barra superior	134
Imagen 85 -	Interfaz – Barra superior sin sesión iniciada	134
Imagen 86 -	Interfaz – Barra superior con sesión iniciada.....	134
Imagen 87 -	Interfaz – Página de registro	136
Imagen 88 -	Interfaz – Formulario de logueo.....	137
Imagen 89 -	Interfaz – Perfil propio	138
Imagen 90 -	Interfaz - Puntuación.....	138
Imagen 91 -	Interfaz – Editar perfil – Información de usuario	140
Imagen 92 -	Interfaz – Perfil de otro usuario	141
Imagen 93 -	Interfaz – Creación de viajes.....	143
Imagen 94 -	Interfaz – Búsqueda de viajes	145
Imagen 95 -	Interfaz – Lista de viajes	146
Imagen 96 -	Interfaz – Información de viaje	147
Imagen 97 -	Interfaz – Solicitud de viaje	149
Imagen 98 -	Interfaz – Búsqueda de paquete	150
Imagen 99 -	Interfaz – Información paquete	151
Imagen 100 -	Interfaz – Creación de paquete.....	152
Imagen 101 -	Interfaz – Módulo viaje.....	154
Imagen 102 -	Interfaz – Módulo solicitud.....	154
Imagen 103 -	Interfaz – Módulo coche	156
Imagen 104 -	Interfaz – Módulo paquete	156
Imagen 105 -	Interfaz – Módulo usuario favorito	157

TABLAS

Tabla 1 - Estimación de costes	46
Tabla 2 - Comparativa MongoDB vs MySQL	118
Tabla 3 - Documentación de API REST	126

1. Introducción

1.1 Motivación

El *consumo colaborativo*^[T1] ha pasado a ser una prioridad para la Unión Europea. Así lo manifiesta el Comité Económico y Social Europeo en un dictamen sobre Consumo Colaborativo o participativo: un modelo de sostenibilidad para el siglo XXI:

“El consumo colaborativo o participativo representa una tercera ola de Internet, donde la gente se encuentra en Internet para compartir fuera de Internet. Y también se describe acertadamente por su función de poner en circulación todo aquello que existe”

Esta tendencia no hace más que confirmarse mes a mes. Sólo en España hay ya contabilizadas más de 300 empresas que prestan servicios de economía colaborativa. En este sector, resultan claves las TIC^[A4] para optimizar la localización de los recursos, agregar grupos de gente desconocida con intereses comunes y construir comunidades.

De los numerosos ámbitos de la vida cotidiana que pueden ser objeto de práctica del consumo colaborativo nos vamos a centrar en este proyecto en el de la movilidad (*carsharing*^[T3], alquiler y uso compartido de vehículos).

Son varias las plataformas que operan desde hace años, con éxito creciente, en este ámbito como veremos en el próximo apartado. No obstante, ninguna de ellas se centra en uno de los colectivos que protagoniza inmensos flujos de movilidad a nivel global: los migrantes. Concretamente, en nuestro caso nos fijaremos en las rutas que conectan Europa con África. A continuación se muestra un mapa que refleja las rutas con un mayor flujo de migrantes entre ambos continentes.

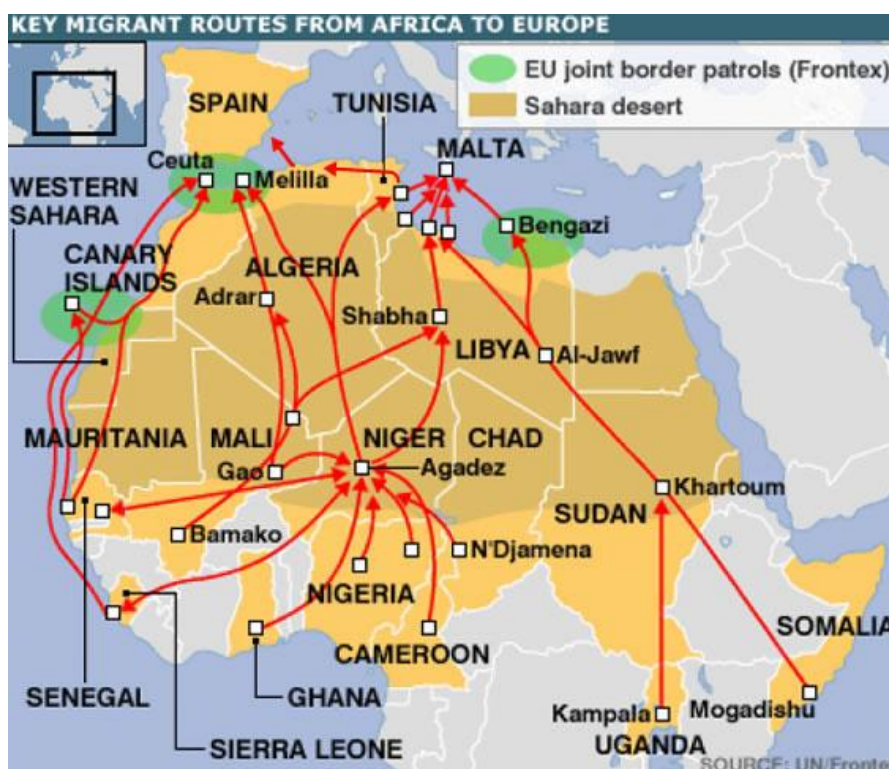


Imagen 1 - Mapa de rutas migrantes Europa-África

(Imagen obtenida de: <http://news.bbc.co.uk/2/hi/europe/6228236.stm>)

Queda evidenciada, por tanto, la conveniencia de crear una plataforma de *economía colaborativa* ^[11] que facilite la movilidad de estos migrantes entre Europa y África, permitiendo crear una comunidad en la que puedan ponerse en contacto fácilmente usuarios con un mismo destino.

Para ello, estudiaremos a continuación algunas de las empresas ya existentes en este ámbito, para marcar sus puntos fuertes y desventajas frente a la aplicación propuesta en este proyecto.

1.2 Estado del arte

Ya sea por motivos de ocio, negocios o personales, en la época en la que vivimos el hecho de desplazarse, tanto en viajes de corta o larga duración, se ha vuelto una necesidad. Sin embargo, no siempre podremos disponer de un vehículo para realizar estos desplazamientos y el transporte público no siempre será lo más conveniente.

Una posible solución a esto es el conocido “*autostop*”, indicando desde el lateral de la carretera a la espera de que algún conductor pare y te acerque a tu destino. Esta solución no es muy efectiva ya que el tiempo de espera puede alargarse mucho; en algunas situaciones o carreteras, incluso, puede ser peligroso la parada de los coches para recoger pasajeros y múltiples problemas más.

Otra solución más aceptable que la anterior consiste en organizar el viaje con personas conocidas que si disponen de vehículo para compartir gastos de transporte. El gran problema de esta solución se halla en que no siempre tendremos conocidos con los que compartir todos los viajes que necesitemos o queramos hacer.

No obstante, podemos tomar como base la idea anterior y, haciendo uso de las tecnologías actuales, crear un sistema mediante el cual, personas desconocidas con origen y destino de viaje comunes pueden ponerse en contacto para realizar el viaje conjuntamente y repartir los gastos de viaje de forma consensuada.

Esta idea ya ha sido desarrollada por múltiples empresas, que actúan de intermediarias y facilitan la comunicación entre personas que desean viajar a cambio de una comisión. Estas empresas utilizan distintos medios para llevar a cabo su negocio, centrándose mayoritariamente en páginas webs o aplicaciones móviles, aprovechando el importante auge que tienen estos años y el amplio sector de personas al que llegan de esta forma.

Puesto que este proyecto estará enfocado de manera similar a lo expuesto anteriormente, deberemos hacer un estudio acerca de las empresas que ya se encuentran actuando en este sector comercial, analizando aquellos puntos que hacen que sus productos tengan éxito y atraigan a los consumidores y también aquellos puntos o necesidades de las que carezcan los productos, para poder crear un artículo que pueda hacer competencia a aquellas que ya se encuentran en mercado.

Aunque existen muchas empresas volcadas en el ámbito del *carsharing*^[13], las mencionadas a continuación son las que más destacan por contar con la mayor comunidad de usuarios o por sus funciones únicas o novedosas, el resto, a día de hoy, cuentan con las mismas funciones a una escala de usuarios menor o más concentrada en algún país en concreto.

1.2.1 BlaBlaCar^{[B1][B2]}

Esta aplicación es el producto más completo y conocido dentro de este sector. Cuenta con una página web y una aplicación de móvil gratuita (tanto en *Android*^[T4] como en *iOS*^[T5]) desde las cuales se pueden acceder a los mismos datos y servicios.

Esta *red social*^[T6] fue creada en 2004 y cuenta con una comunidad de más de 20 millones de usuarios y con una ocupación media de 2.8 personas por vehículo que ofrece un viaje, lo que la hace una aplicación muy deseable para encontrar compañero de viaje de forma rápida.

A continuación analizaremos la aplicación por partes, discutiendo aciertos y fallos de esta:

1.2.1.1 El sistema en general

Aciertos

- ✓ La aplicación mantiene una gran comunidad de usuarios, creando un sistema de ranking y opiniones en la que todos los usuarios participan. Esto incentiva el utilizar repetidas veces la aplicación ya que crea lazos entre los usuarios y posiciones privilegiadas que se alcanzan con tiempo, uso de la aplicación y opiniones del resto de la comunidad.
- ✓ Una gran ventaja de esta aplicación es la posibilidad de aceptar o denegar quién puede acompañarte, valiéndose de un sistema de puntuaciones con opiniones de viajes anteriores con otros usuarios. Esto crea seguridad en la persona, que puede hacerse una idea anticipada de con quién va a viajar.
- ✓ La empresa pone gran empeño en proporcionar toda la ayuda necesaria para resolver las dudas de los usuarios. Para ello cuenta con un listado de preguntas frecuentes, tutoriales sobre las acciones básicas (como cómo crear un viaje o apuntarse a uno ya creado) y un gran grupo de personas trabajando en atención al cliente, que aseguran una respuesta en menos de 24 horas en caso de que tu pregunta no se encuentre dentro de la lista nombrada anteriormente.
- ✓ Cuenta con una web para cada uno de los países en los que se encuentra disponible, en el idioma oficial de ese país. Esto permite reducir el número de resultados mostrados, enseñando solo aquellos con origen en el país en el que se encuentra el usuario. Además, al encontrarse en la lengua nativa del usuario y no en una lengua internacional como el Inglés permite llegar a un mayor número de usuarios y facilita la tarea al usuario.
- ✓ Se encuentra disponible para viajes nacionales en 19 países actualmente.

Fallos

✗ Aunque permite los viajes internacionales entre cualquier par de países, estos viajes son escasos y se centran en ir entre las dos capitales de los países, lo que limita los viajes fuera del país.

✗ No contempla la posibilidad de realizar trasbordos para llegar a un destino, por lo que, en caso de no existir un viaje directo, es el usuario el que debe buscar los viajes por separado, contemplando todas las posibilidades de manera individual, haciendo de esta una tarea ardua e incómoda.

1.2.1.2 Página web

Esta sección, debido a su amplitud la separaremos a su vez en varias subsecciones, para analizar con más detalle cada una de ellas:

1.2.1.2.1 General

Aciertos

✓ Tiene una amplia documentación en todas las páginas de su web acerca de cómo realizar cualquier actividad, facilitando la tarea del usuario y resolviendo las dudas del usuario dudas de forma rápida.

✓ Te permite en todo momento cambiar la divisa con la que deben mostrarse los precios.

✓ La página de ayuda es muy simple y útil, proporcionando al usuario de forma fácil, amigable y rápida, la ayuda necesaria para usar el servicio y la página web y resolver cualquier duda acerca de su funcionamiento.

Fallos

✗ No cuenta con una web responsiva, lo que puede resultar al usuario incómodo si quiere visitar la web desde un dispositivo de pantalla pequeña, como un *Smartphone* o *tableta*.

1.2.1.2.2 Registro y logueo

Aciertos

- ✓ Cuenta con un menú superior que te permite acceder al formulario de registro o logueo desde cualquier página de la web.
- ✓ El formulario de registro, en una página individual, es muy simple y básico, pidiendo únicamente los datos pertinentes para el uso del servicio propuesto.
- ✓ Permite un registro mediante la red social^[T6] *Facebook*, permitiendo al usuario aunar sus cuentas y simplificar y acelerar el proceso de registro.
- ✓ El formulario de inicio de sesión se abre como un módulo dinámico que evita al usuario tener que visitar otra página para loguearse.

Fallos

- ✗ Cuando te encuentras en la página de registro este menú desaparece, lo que impide al usuario volver a la página principal, excepto que se recargue la página.
- ✗ El formulario no está centrado, sino en un lateral, resaltando lo mismo, o incluso menos, que los enlaces de ayuda.
- ✗ Por defecto, la opción de suscripción a noticias está activada, lo que puede suponer correo no deseado para el usuario.
- ✗ El formulario de registro se encuentra en una página aparte, al contrario que el de inicio de sesión, lo que obliga al usuario a cambiar de página para registrarse, perdiendo la página en la que se encuentre el usuario en ese momento.
- ✗ No permite el registro desde otras redes sociales^[T6] que no sea *Facebook*, como puede ser *Google+*.

1.2.1.2.3 Navegación

Aciertos

- ✓ Cuenta con un estilo gráfico limpio y simple, adornado con efectos suaves, lo que hace que la utilización de la aplicación sea fácil y cómoda para el usuario, que no tiene que esforzarse en entender o buscar cómo realizar aquello que desea.
- ✓ La página principal destaca claramente los contenidos importantes, permitiendo a los usuarios un acceso claro y rápido sin esfuerzo.

Fallos

- ✗ No utiliza la tecnología **AJAX**^[A1], obligando a recargar la página en cada nueva interacción, aumentando la carga de red que el usuario debe soportar y aumentando los tiempos de carga medios.
- ✗ La opción de "**Like**" de *Facebook* desaparece al haber iniciado sesión (lo cual no tiene del todo sentido al ser una opción normalmente asociada al conocimiento de la página y, por lo tanto, su uso estando logueado).

1.2.1.2.4 Aplicación móvil

Con más de 5 millones de descargas, esta aplicación permite acceder a **Blablacar** cómodamente desde un dispositivo *Smartphone* o *tableta*.

Aciertos

- ✓ Nada más iniciar la aplicación te permite seleccionar el país de procedencia, para mostrar resultados solo relacionados con este.
- ✓ Cuenta con una interfaz limpia y simple que facilita la interacción con el usuario.
- ✓ No cuenta con publicidad.

Fallos

- ✗ No encontrados.

1.2.2 Amovens^{[B3][B4]}

Esta aplicación española fue lanzada al mercado en el año 2009. Tras un crecimiento exponencial en 2014, pasa a formar parte de la red de coche compartido **GoMore** —una plataforma de origen escandinavo presente en varios países de Europa— con el objetivo de acelerar su progresión y ofrecer una gama más completa de servicios de movilidad a sus usuarios y fomentar un uso más práctico y eficiente del coche.

Esta aplicación es completamente gratuita para un usuario particular, por lo que no aumenta el coste que debe pagar el usuario para viajar.

1.2.2.1 Servicio especial para empresas

Amovens cuenta también con un servicio especial para empresas, para que estas fomenten que sus empleados compartan el transporte. Es lo que llaman “carpooling”. Es un servicio especialmente pensado para empresas grandes, con muchos empleados, el sistema se encarga de buscar las rutas compatibles según el lugar de residencia de cada uno de ellos. Para estas empresas el precio de uso es de 199€/mes, reduciéndose a 99€/mes si se trata de una **PYME** u **ONG**.

Algunas de las ventajas que aporta este servicio para empresas son las siguientes:

- ✓ Reducción de la necesidad de plazas de aparcamiento
- ✓ Reducción del estrés debido a la disminución de la cantidad de tráfico
- ✓ Menor contaminación
- ✓ Ahorro económico en combustible y mantenimiento
- ✓ Mejora de las relaciones interpersonales

1.2.2.2 Aplicación web

Junto con la página web, Amovens ofrece una aplicación de móvil asociada a esta, desde la cual puedes realizar los mismos procesos que desde la página web.

- ✓ Cuenta con una interfaz clara e intuitiva. con la información justa y necesaria para buscar y crear viajes, facilitando la tarea al usuario.
- ✓ No cuenta con publicidad.

Fallos

- ✗ No encontrados.

1.2.2.3 Servicio de alquiler de coches

Como una nueva y novedosa funcionalidad de la aplicación, **Amovens** permite el alquiler de coches entre usuarios, de este modo los propietarios pueden conseguir ingresos (hasta 3.000€ al año) mientras que los arrendatarios pueden acceder a una amplia variedad de vehículos de forma económica y flexible, cerca de donde se encuentran.

Amovens se encarga de gestionar la relación entre ambas partes y de aportar un seguro especializado a todo riesgo, que sustituye al del propietario durante el periodo de alquiler. También asegura la asistencia en carretera.

Además, la empresa ofrece un servicio de alquiler de coches compartidos que, por una cuota de entre 295 o 363 euros mensual (295€ si se arrenda por 6 meses, 363€ si se hace por 1 año), permite a un usuario alquilar un vehículo para un período de al menos seis meses.

En esas cuotas está incluido el seguro del automóvil y la asistencia en carretera, explica la empresa en un comunicado.

Para este nuevo servicio, **Amovens** tiene un acuerdo de colaboración con **LeasePlan**, empresa mundial de renting de vehículos y gestión de flotas.

1.2.3 RoadSharing^[B5]

Esta aplicación italiana es utilizada internacionalmente y se encuentra disponible en 5 idiomas (Italiano, Español, Inglés, Francés y Alemán).

Al igual que las anteriores, permite compartir coche a personas con destinos comunes.

Esta página cuenta, no obstante, con algunas desventajas frente a las anteriores:

- ✖ No permite la comunicación entre usuarios por medio telefónico, solo por la mensajería interna de la aplicación.
- ✖ No cuenta con una aplicación móvil asociada a la página web.
- ✖ No cuenta con una página web responsiva^[17] que pueda ser utilizada en pantallas pequeñas, pudiendo suplantar la carencia de aplicación móvil.
- ✖ Cuenta con una comunidad mucho más pequeña de usuarios y centrada mayoritariamente en Italia.

1.2.3.1 Sistema de eventos

Como novedad, esta página web añade un sistema de eventos, en el cual se muestran eventos, convenciones, festividades, etc. de distintos países, y se pueden crear o unirse a viajes hacia estos eventos. Esto permite que los usuarios con aficiones comunes se conozcan y puedan volver a compartir viaje para estos eventos.

1.2.4 Conclusiones

Observando las ventajas y desventajas de los servicios similares ya en mercado a nuestro proyecto, podemos ver aquellos requisitos o mejoras opcionales que harían destacar a nuestra aplicación frente al resto. Teniendo en cuenta entonces estas nuestra aplicación debe tener:

- Ausencia de publicidad externa dentro de la aplicación.
- Conseguir una amplia comunidad activa que mantenga viva la página.
- Centrar el uso de la aplicación en regiones donde no exista una aplicación ya en mercado establecida y con una gran actividad.
- Contar con una aplicación móvil enlazada con la aplicación web o, en caso de no ser posible, contar con una página web responsiva^[17] que pueda ser visualizada de forma cómoda en pantallas de poca resolución.
- Contar con un servicio novedoso o poco conocido que no usen las aplicaciones similares.
- Mantener una interfaz limpia, clara y concisa, sin sobrecargar y que permita al usuario encontrar rápidamente aquello que busca, eliminando elementos innecesarios que proveen otras aplicaciones centradas en estos.
- Realizar la mayor parte del procesado en la parte del cliente para agilizar el proceso y proveer de una navegación fluida al usuario, con el menor número de esperas posibles.

1.3 Estructura del documento

A lo largo de todo el documento se encontrarán referencias bibliográficas, las cuales se representarán como ^[BX], sustituyendo **X** por el número de referencia que se encontrará en el apartado bibliografía. Esto se encontrará al final del título del apartado para el cual ha sido necesaria esa referencia.

De igual manera, se hará referencia al apartado glosario al final del documento, situando ^[TX] al final de una palabra o grupo de palabras y ^[AX] para los acrónimos. Al igual que antes, se sustituye **X** por el índice dentro de los términos o acrónimos, donde están explicados estos.

Este documento está compuesto por las siguientes partes:

- **Resumen:** Breve resumen de la funcionalidad del proyecto, así como un resumen más extendido de ésta en inglés.
- **Introducción:** Pequeña introducción al proyecto, tratando los siguientes temas:
 - Motivación que ha llevado a la creación del proyecto. .
 - Estado del arte analizando las aplicaciones similares existentes en el mercado actual.
 - Estructura básica del documento.
 - Lista de objetivos a lograr durante la realización del proyecto
- **Planificación:** Exposición de la planificación del proyecto, las distintas etapas que está compuesto y una estimación de presupuesto del mismo.
- **Análisis:** Fase en la que se especifican los requisitos del sistema y se hace un estudio de los casos de uso y flujos del sistema.
- **Diseño:** Fase en la que se crea una estructuración y diseño inicial de las clases necesarias para el programa y la interfaz de usuario^[18].
- **Implementación:** Aspectos más relevantes de la implementación del proyecto y problemas ocurridos durante esta, junto con las soluciones aplicadas, separados en los siguientes apartados:
 - Herramientas utilizadas, donde se exponen y explican las diversas herramientas empleadas para la creación del proyecto.
 - Fases del desarrollo, donde se explican los diversos pasos seguidos en la implementación, analizando los problemas encontrados y las soluciones propuestas, así como las herramientas utilizadas en cada caso.
- **Diseño de interfaz de usuario:** Donde se muestra todas las pantallas y módulos de la interfaz y su utilidad y funcionamiento.
- **Conclusiones:** Resumen final del proyecto explicando las conclusiones

obtenidas tras la creación de este y posibles vías futuras a las que encaminar el proyecto.

- **Bibliografía:** Lista de fuentes de información utilizadas para la realización del proyecto.
- **Anexos:** Otros documentos de valor a la hora de realizar el proyecto, adjuntados en la propia memoria, así como un glosario de términos y acrónimos.

1.4 Objetivos del proyecto

Para cubrir las necesidades que han llevado a la creación de este proyecto se necesitan una serie de objetivos. Unos serán obligatorios para el correcto funcionamiento de la aplicación, y otros serán opcionales, que facilitarán la tarea del usuario y realzarán la funcionalidad del programa. A continuación se muestra una lista de aquellos objetivos a lograr con el proyecto, los cuales volverán a ser tratados al final de la memoria, comprobando si se han cumplido y, en caso de no ser así, por qué.

Objetivo 1	<i>Identificación única de usuario</i>
Tipo	Obligatorio
Descripción	Obliga a un usuario a identificarse para poder interactuar con la aplicación

Objetivo 2	<i>Creación de viajes</i>
Tipo	Obligatorio
Descripción	Permite a un usuario dar los datos acerca del viaje que va a realizar para permitir a otros usuarios unirse a él.

Objetivo 3	<i>Búsqueda de viajes según criterios</i>
Tipo	Obligatorio
Descripción	Permite a un usuario encontrar los viajes que se adaptan a sus necesidades

Objetivo 4	<i>Solicitud de participación en viajes</i>
Tipo	Obligatorio
Descripción	Permite a un usuario solicitar ir de pasajero en un viaje

Objetivo 5	<i>Solicitud de transporte de paquetes en viajes</i>
Tipo	Obligatorio
Descripción	Permite a un usuario solicitar que un paquete de su propiedad vaya como equipaje en un viaje

Objetivo 6	<i>Sistema de trato de solicitudes</i>
Tipo	Opcional
Descripción	Permite a un usuario decidir si aceptar o no las solicitudes realizadas por otros usuarios para su viaje. (En caso de no realizarse este objetivo las solicitudes serían automáticamente aceptadas)

Objetivo 7	<i>Sistema de puntuaciones de usuarios y viajes</i>
Tipo	Obligatorio
Descripción	Permite a un usuario que ha ido de pasajero en un viaje puntuar el mismo y a su conductor, creando un sistema de puntuaciones de usuario que se mostrará públicamente.

Objetivo 8	<i>Control de localización de paquetes</i>
Tipo	Opcional
Descripción	Permite a un usuario tener constancia de la localización aproximada de su paquete mientras este se encuentra en el trayecto de un viaje.

Objetivo 9	<i>Sistema de mensajería interna</i>
Tipo	Opcional
Descripción	Permite crear un sistema de mensajería por la cual los usuarios pueden de forma privada comunicarse entre sí dentro de la misma aplicación.

Objetivo 10	<i>Soporte de idiomas</i>
Tipo	Opcional
Descripción	Separar el texto mostrado en la aplicación, permitiendo una fácil transcripción de esta a otros idiomas.

2. Planificación

2.1 Fases

La planificación realizada para el desarrollo del proyecto se dividió en varias partes, para poder abarcar correctamente los distintos campos de acción del proyecto:

2.1.1 Fase inicial

Partimos de la necesidad de crear una aplicación para facilitar la comunicación entre usuarios que necesitan realizar un viaje entre España y Senegal. Esta necesidad ha sido planteada por la empresa “*Acento Comunicación*”^[B6] y se creó este proyecto como solución a esta necesidad, esbozando, con ayuda de *Rosana Montes Soldado* y el equipo de trabajo de la empresa, se esbozó una idea y requisitos generales que debía cumplir la aplicación.

Tras una serie de reuniones se decidió usar *Polymer*^[5.1.2.2] como base para la página web, por su gran modularidad, flexibilidad y escalabilidad, desarrollando la página web con la tecnología puntera de la que se sirven grandes empresas como Google.

Como servidor que mantendrá la página se decidió usar *NodeJS*^[5.1.3.1] junto con Express, para usar una *API REST*^[A3] que, sin cambio alguno, permita usarse tanto en la aplicación web como en una aplicación móvil.

Para el almacenamiento de datos se barajó la idea de usar una base de datos *MongoDB* por sus mayores facilidades y ventajas para el proyecto, pero la empresa propuso *MySQL*^[5.1.1.2], ya que tenían una base de datos de este tipo ya implantada en su empresa, por lo que al final se optó por esta segunda.

Aunque la empresa pidió una aplicación sin una interfaz detallada (ya que tenían a sus propios diseñadores dentro de la empresa), se realizó un estudio y estructuración básicos para las distintas páginas que mostraría la aplicación.

2.1.2 Fase de análisis

En esta etapa se estudió aquellos requisitos básicos para el correcto funcionamiento de la aplicación, así como los recursos, tanto software como hardware, necesarios para cumplir estos requisitos.

2.1.2.1 Especificación de requisitos

Se realizó un estudio de todos los requisitos que debería cumplir el programa, así como de aquellos requisitos opcionales que le aportarían un valor añadido al mismo.

2.1.2.2 Recursos necesarios

Se analizaron las distintas bibliotecas y recursos disponibles y se seleccionaron aquellas que encajaban mejor con los requisitos del sistema.

2.1.3 Fase de desarrollo

Esta fase ha sido la más extensa de todas, en la que se ha dado forma al programa en sí.

La fase se dividió en pequeños hitos en los que se iban resolviendo uno por uno los requisitos del programa. Estos hitos fueron:

- Diseño de la base de datos
- Creación de la *API REST*^[A3]
- Estructuración de la aplicación web
- Sistema de sesiones
- Conexión con la *API*^[A5] de Google

Para lograr estos hitos, fue necesario el estudio y adquisición de conocimientos en las diversas herramientas a tratar a lo largo del proyecto. Más en concreto, fue necesario un estudio adicional en los siguientes apartados:

- *Polymer*^[B9]

Necesario para la creación de la estructura de la aplicación web, es una biblioteca aún en desarrollo que sacó su versión estable hace unos pocos meses.

Se comenzó su estudio tratando con su versión *alfa 0.5* y después tuvo que adaptarse los conocimientos aprendidos con modificaciones al pasar a la versión estable 1.0.

Ya que es una tecnología aun en desarrollo, su documentación no está aún completa, por lo que existen algunos módulos y elementos sin documentar por los que hubo que preguntar en sus foros oficiales para descubrir su correcto funcionamiento.

Una vez entendidos los módulos que iban a poder utilizarse en la web, hubo que analizar cómo enlazarlos con la obtención dinámica de datos por parte del servidor y crear funciones en *JavaScript*^[5.1.1.1] para el correcto funcionamiento tanto de los módulos ya creados como los módulos creados específicamente para la aplicación.

Por último, fue necesario estudiar la relación de *Polymer* con la hoja de estilo *CSS*^[5.1.1.4], pudiendo ser independiente para cada módulo o tener centralizados en una misma hoja aquellos aspectos comunes que tendrán los módulos y la web, para mantener un equilibrio entre todos los elementos de la página, manteniendo un diseño común.

- **API REST^[B8]**

A la hora de interactuar con la base de datos se optó por la creación desde 0 de una *API REST^[A3]* específica para la aplicación. Esto permitiría que pudiera ser usada tanto en la aplicación web como en otras aplicaciones futuras de móvil o de escritorio sin realizar ningún cambio.

Para la creación de esta *API^[A5]* hubo que utilizar diversos programas para asegurar en cada paso que se realizaba correctamente. Para ello se tuvo que estudiar y comprender las siguientes tecnologías:

- **MySQL Workbench:** Permite comprobar las sentencias *SQL^[5.1.1.2]* que se van a realizar en la base de datos. Esto permitirá asegurarse de cada secuencia que va a tener que realizar la *API^[A5]*.
- **NodeJS con Express:** Servidor que permite la creación de la *API REST^[A3]* y trata con los datos almacenados en la base de datos. Existe una amplia documentación sobre este proceso.
- **Postman:** Permite comprobar que cada *URL* se ha ejecutado correctamente y devuelve los datos adecuados.

- **API de Google^{[B14] [B15] [B16]}**

Para la obtención de precios y rutas necesarias para la aplicación, además de mostrarlas gráficamente se ha optado por utilizar la *API^[A5]* de mapas que proporciona Google. Con esta *API^[A5]* se van a poder realizar varios procesos necesarios para la aplicación como son los siguientes:

- **Google Directions:** Obtención de los kilómetros que cubre una ruta, para obtener, a partir de ellos, el coste del viaje por cada tramo de la ruta. También permite obtener el tiempo aproximado de duración del trayecto, permitiendo estimar tiempos de llegada y parada en los lugares determinados.
- **Google Maps:** Obtención del mapa con la ruta marcada en ella, para poder mostrar visualmente el trayecto a recorrer.
- **GeoCoding** Obtención de la latitud y longitud, así como otros datos asociados a una localización, para su posterior uso en el mapa u otras tareas.

2.1.4 Fase de prueba y corrección

Enlazada con la fase anterior, en la finalización de cada hito se realizaba una prueba de funcionamiento, solucionando los errores que ocurrieran.

2.1.5 Redacción de la memoria

Durante el tiempo que duraron todas las fases anteriores se hizo una recogida de información y datos que, llegando a los últimos hitos a completar del desarrollo se plasmaron y ordenaron correctamente, para conformar la memoria actual.

El proyecto comenzó en Junio, abarcando 3 meses hasta su finalización, este periodo de tiempo ha estado dividido en las fases expuestas anteriormente según el siguiente diagrama de Gannt:

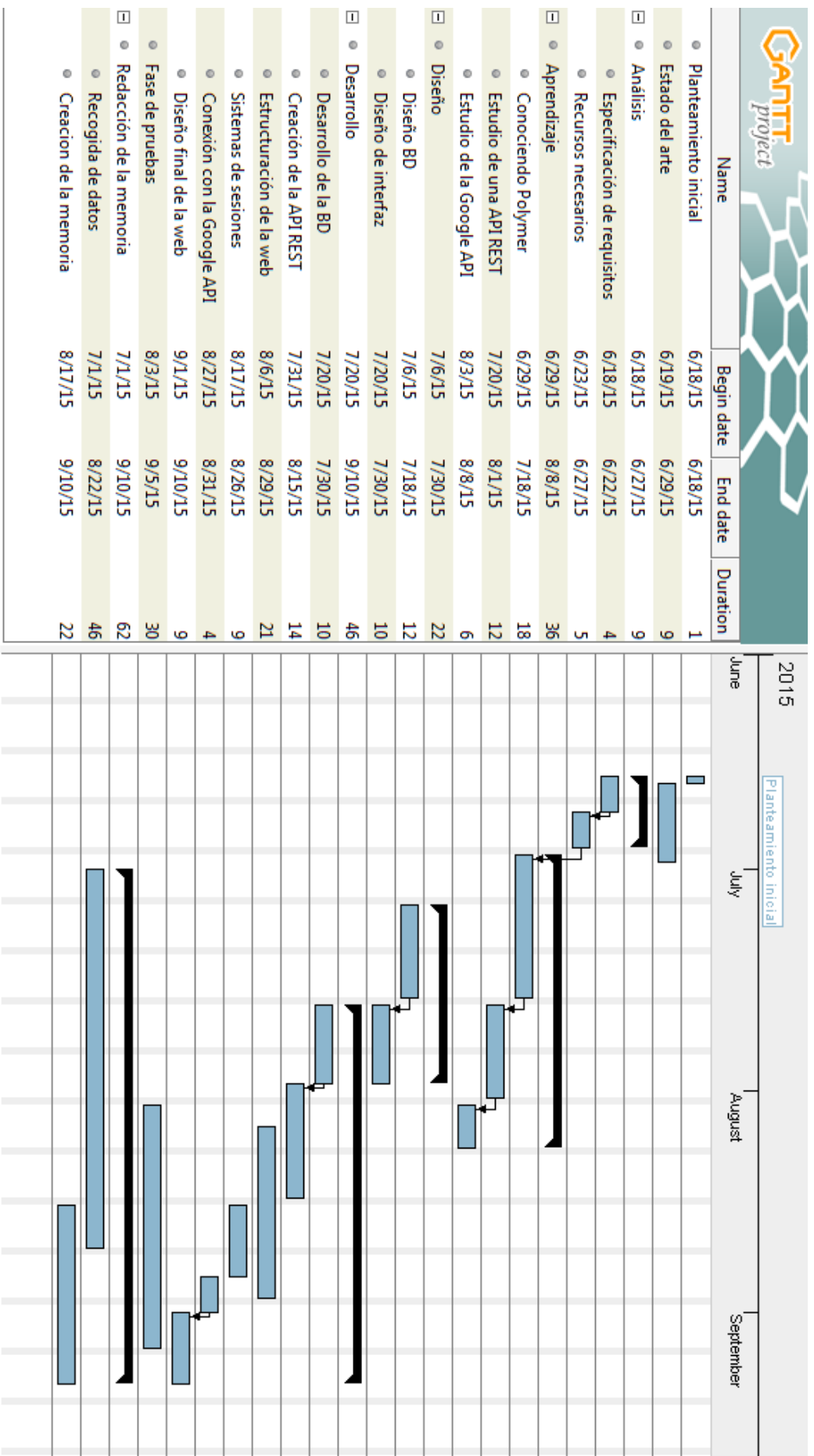


Imagen 2 - Diagrama de Gantt

2.2 Presupuesto

2.2.1 Recursos

En este apartado se van a listar todos los recursos inventariables de hardware y software, así como las herramientas utilizadas y los lenguajes de programación.

2.2.1.1 Hardware

- Laptop donde se han realizado el proyecto: *HP ProBook 6460b Intel® Core™ i5-2450M CPU - @ 2.5GHz.*

2.2.1.2 Software

- **Sistema Operativo:** *Ubuntu 14.04 LTS 64 bits*
- **Lenguajes de programación:**
 - *Javascript*
 - *SQL (MySQL)*
- **Tecnologías de desarrollo:**
 - *AJAX*
- **Bibliotecas**
 - *Polymer 1.0*
- **Lenguajes de etiquetado:**
 - *HTML5*
- **Lenguaje de estilo:**
 - *CSS3*
- **Diseño de diagrama UML:** *Plugin easyUML*
- **Editor de código:** *Sublime Text*
- **Administración de BD:** *MySQL WorkBench*
- **Testeo de API REST:** *Postman*
- **Control de versiones:** *GIT*
- **Servidor Web y de API REST:** *NodeJS con express 4*
- **Procesador de texto:** *Microsoft Word*
- **Diseño de diagrama de Gantt:** *GanttProject*

2.2.2 Costes

2.2.2.1 Licencias

Un apartado a tener en cuenta a la hora de estimar costes de un proyecto son las licencias asociadas a los recursos usados, por ello se ha elaborado una lista con los recursos y sus licencias asociadas, con un enlace a la explicación detallada de cada una:

- **MySQL** ^[B19]
 - Licencia GPL o uso comercial (GPL en este caso)
 - <http://www.gnu.org/licenses/gpl.html>
- **Polymer** ^[B20]
 - Licencia BSD
 - <http://polymer.github.io/LICENSE.txt>
- **NodeJS con Express** ^[B21]
 - Licencia MIT
 - <http://opensource.org/licenses/mit-license.php>
- **MySQL Workbench** ^[B19]
 - Licencia GPL
 - <http://www.gnu.org/licenses/gpl.html>
- **Postman**
 - Gratuito

Teniendo en cuenta estas licencias, el coste por el uso de software utilizado en el proyecto es de 0€.

2.2.2.2 Recursos materiales

En recursos materiales se tendrá en cuenta el laptop usado para el proyecto.

Los ordenadores suelen tener un período de amortización estipulado entre 2 y 4 años. Para este caso se hará una media y se estipulará un período de 3 años de amortización.

El equipo costó 800€, por lo tanto, se amortizará 267€ al año. Teniendo en cuenta que la duración del proyecto es de 3 meses, el coste final de los recursos materiales asciende a 66€.

2.2.2.3 Recursos humanos ^[B18]

Para el cálculo de costes de personal se ha tenido en cuenta un único empleado Programador Senior.

Tomando datos obtenidos para el trabajo anterior en *InfoJobs*, este cobra entre 23.000 y 26.000€ anuales. Teniendo esto en cuenta se hace la correspondencia para el tiempo de desarrollo del proyecto.

Puesto que han sido 3 meses y un trabajador cobra unos 2.042 € mensuales, se estima un salario de 6.126 €.

2.2.2.4 Otros

Dentro de este apartado se engloban costes indirectos como material de mesa, conexión a Internet, luz, gastos de transporte, etc. Su coste suele rondar el 10% del gasto del personal, por lo que se estima un coste aproximado de 612,6€.

2.2.2.5 Total

A continuación se muestra una tabla resumen con los datos de costes obtenidos en los apartados anteriores, haciendo una estimación del coste mensual del proyecto y su coste total, tanto global como para cada apartado.

Descripción	Coste mensual	Coste total
Licencias de software	0 €	0 €
Estación de trabajo	22,4 €	67,2 €
Programador Senior	2.042 €	6.126 €
Otros	204,2 €	612,6 €
Total	2.268,6 €	6.805,8 €

Tabla 1 - Estimación de costes

3. Análisis

3.1 Especificación de requisitos

En esta sección se detallan las distintas condiciones y necesidades que ha de satisfacer nuestro producto. Seguiremos para ello un esquema que nos permita describir los requisitos de una forma clara y concisa.

Todos los requisitos se identifican de forma inequívoca mediante un código que constará de una codificación indicando el tipo de requisito y un número de orden. Este código será utilizado como referencia cada vez que sea necesario mencionarlo a lo largo del desarrollo del proyecto.

3.1.1 Requisitos funcionales

Para que el programa a crear sea funcional y cumpla las necesidades del usuario, debe cumplir una serie de requisitos. Estos requisitos se pueden dividir en grupos según su función:

- El primero de ellos relacionado con el tratamiento de una sesión de usuario.
- En el segundo grupo se encuentran aquellos requisitos relacionados con el tratamiento de viajes.
- En el tercer grupo situamos aquellos requisitos encargados de tratar con las solicitudes.
- En el cuarto grupo se encuentran aquellos requisitos encargados de tratar con paquetes.
- En el quinto grupo se hallan los requisitos relacionados con el tratamiento de coches.
- En el sexto grupo se agrupan los requisitos que tratan con usuarios.
- Por último, en el séptimo grupo se encuentran aquellos requisitos que tratan con notificaciones.

Siguiendo la guía anterior, agruparemos los requisitos creados en estos grupos de la manera que se muestra a continuación.

3.1.1.1 Sesión

RF1	<i>Registro</i>
Explicación	Registra a un usuario en la aplicación.
Datos de entrada	Información del nuevo usuario.
Datos de salida	Genera un nuevo usuario en la BD ^[A6] con los datos introducidos.

RF2	<i>Inicio de sesión</i>
Explicación	Permite a un usuario iniciar sesión dentro de la aplicación.
Datos de entrada	Correo y contraseña del usuario a loguearse.
Datos de salida	Sesión iniciada del usuario.

RF3	<i>Cerrar sesión</i>
Explicación	Termina la sesión de un usuario dentro de la aplicación.
Datos de entrada	ID ^[A7] del usuario a cerrar sesión.
Datos de salida	Ninguno.

RF4	<i>Editar perfil</i>
Explicación	Permite modificar algunos datos del perfil del usuario que tiene iniciada la sesión.
Datos de entrada	Datos modificados del usuario.
Datos de salida	Guarda la información actualizada en el sistema.

3.1.1.2 Viajes

RF5	<i>Ver viaje</i>
Explicación	Muestra información detallada acerca de un viaje.
Datos de entrada	Identificador del viaje a mostrar.
Datos de salida	Interfaz con los datos del viaje.

RF6	<i>Buscar viajes</i>
Explicación	Muestra una lista de viajes que se atienen a unos criterios de búsqueda especificados.
Datos de entrada	Criterios de búsqueda de viajes.
Datos de salida	Lista de viajes encontrados que concuerdan con la búsqueda.

RF7	<i>Crear viaje</i>
Explicación	Crea un viaje con los datos indicados.
Datos de entrada	Datos del viaje a crear e identificador del usuario que lo ha creado.
Datos de salida	Crea un nuevo viaje asociado al usuario indicado en la BD ^[A6] del sistema.

RF	<i>Actualizar viaje</i>
Explicación	Un usuario puede cambiar algún parámetro del viaje que ha creado.
Datos de entrada	Identificador del usuario y del viaje a modificar y datos modificados del viaje.
Datos de salida	Actualiza la información del viaje en la BD ^[A6] .

RF9	<i>Puntuar viaje ajeno</i>
Explicación	Permite a un usuario puntuar un viaje del cual ha sido pasajero.
Datos de entrada	Identificador del usuario que hace la puntuación y del viaje al que se hace. Puntuaciones y comentario.
Datos de salida	Crea en la BD ^[A6] un nuevo comentario relacionado con un viaje y un usuario.

RF10	<i>Ver lista de viajes</i>
Explicación	Muestra una lista de viajes pertenecientes a un usuario.
Datos de entrada	Identificador del usuario.
Datos de salida	Lista de viajes que ha creado el usuario y a los que va a viajar como pasajero y aún están activos.

3.1.1.3 Solicitud

RF11	<i>Crear solicitud</i>
Explicación	Crea una nueva solicitud para un viaje.
Datos de entrada	Identificador del usuario que realiza la solicitud y del viaje al que la realiza.
Datos de salida	Crea la solicitud en la BD ^[A6] y crea una notificación al usuario que ha creado el viaje especificado.

RF12	<i>Borrar solicitud propia</i>
Explicación	Elimina una solicitud que ha creado el mismo usuario anteriormente.
Datos de entrada	Identificador de la solicitud a borrar.
Datos de salida	Elimina de la BD ^[A6] del sistema la solicitud.

RF13	<i>Aceptar solicitud ajena</i>
Explicación	Acepta una solicitud de un usuario para viajar en el viaje que ha creado el usuario que la acepta.
Datos de entrada	Identificador de solicitud, viaje y usuario que ha creado la solicitud.
Datos de salida	Actualiza la solicitud en la BD ^[A6] y crea una notificación para el usuario que creó la solicitud.

RF14	<i>Rechazar solicitud ajena</i>
Explicación	Rechaza una solicitud de un usuario para viajar en el viaje que ha creado el usuario que la rechaza.
Datos de entrada	Identificador de solicitud, viaje y usuario que ha creado la solicitud.
Datos de salida	Actualiza la solicitud en la BD ^[A6] y crea una notificación para el usuario que creó la solicitud.

RF15	<i>Ver lista de solicitudes</i>
Explicación	Muestra una lista de solicitudes que tiene un usuario.
Datos de entrada	Identificador del usuario.
Datos de salida	Lista de solicitudes pertenecientes al usuario indicado.

3.1.1.4 Paquetes

RF16	<i>Ver paquete</i>
Explicación	Muestra información detallada acerca de un paquete.
Datos de entrada	Identificador del paquete.
Datos de salida	Interfaz mostrando toda la información acerca del paquete indicado.

RF17	<i>Crear paquete</i>
Explicación	Crea un nuevo paquete con los datos indicados.
Datos de entrada	Identificador del usuario que crea el paquete y datos del paquete.
Datos de salida	Crea en la BD ^[A6] del sistema un nuevo paquete y genera un identificador para este.

RF18	<i>Borrar paquete</i>
Explicación	Elimina un paquete.
Datos de entrada	Identificador del paquete.
Datos de salida	Elimina de la BD ^[A6] del sistema el paquete indicado.

RF19	<i>Actualizar estado de paquete</i>
Explicación	Actualiza la información de estado de un paquete.
Datos de entrada	Identificador del paquete y nuevo estado de este.
Datos de salida	Actualiza la información del paquete dentro de la BD ^[A6] del sistema.

RF20	<i>Ver lista de paquetes</i>
Explicación	Muestra una lista de paquetes que ha creado un usuario.
Datos de entrada	Identificador del usuario.
Datos de salida	Lista de paquetes pertenecientes a un usuario.

3.1.1.5 Coches

RF21	<i>Crear coche</i>
Explicación	Crea un nuevo coche asociado a un usuario.
Datos de entrada	Identificador del usuario que crea el coche y datos del coche.
Datos de salida	Crea en la BD ^[A6] un nuevo coche en caso de no existir y añade al usuario indicado a la lista de conductores de este

RF22	<i>Borrar coche</i>
Explicación	Elimina al usuario como conductor de un coche.
Datos de entrada	Identificador de usuario y coche.
Datos de salida	Elimina al usuario como conductor del coche indicado y, en caso de no quedar ningún conductor, elimina también el coche de la BD ^[A6] del sistema.

RF23	<i>Ver lista de coches</i>
Explicación	Muestra una lista de coches de los que el usuario es conductor.
Datos de entrada	Identificador del usuario.
Datos de salida	Lista de coches de los que el usuario es conductor.

3.1.1.6 Usuarios

RF24	<i>Ver perfil de usuario</i>
Explicación	Muestra la información de un usuario.
Datos de entrada	Identificador de un usuario.
Datos de salida	Interfaz con la información pública de un usuario y llama al RF10 .

RF25	<i>Ver lista de favoritos</i>
Explicación	Muestra la lista de usuarios agregados como favoritos por otro usuario.
Datos de entrada	Identificador de usuario.
Datos de salida	Lista de usuarios favoritos del usuario indicado.

RF26	<i>Añadir favorito</i>
Explicación	Añade como favorito a un usuario.
Datos de entrada	Identificador del usuario que pide el favorito y del que ha sido agregado como tal.
Datos de salida	Crea en la BD ^[A6] la relación de favorito entre el usuario y su favorito y se llama al RF29 .

RF27	<i>Eliminar favorito</i>
Explicación	Elimina de favoritos a un usuario.
Datos de entrada	Identificador del usuario que quiere eliminar el favorito y del que ha sido agregado como tal.
Datos de salida	Se elimina de la BD ^[A6] la relación de favorito.

3.1.1.7 Notificaciones

RF28	<i>Leer notificación</i>
Explicación	Marca una notificación como leída.
Datos de entrada	Identificador de la notificación.
Datos de salida	Actualiza la información de la notificación en la BD ^[A6] del sistema para marcarla como leída.

RF29	<i>Crear notificación</i>
Explicación	El sistema crea una notificación.
Datos de entrada	Identificador de usuario al que se le añadirá la notificación y datos de esta.
Datos de salida	Crea una nueva notificación en la BD ^[A6] del sistema con los datos introducidos y relacionada con el usuario indicado.

RF30	<i>Borrar notificación</i>
Explicación	Elimina una notificación.
Datos de entrada	Identificador de la notificación.
Datos de salida	Elimina de la BD ^[A6] la notificación.

3.1.2 Restricciones semánticas

En este apartado se muestran aquellas restricciones semánticas que afectan a los requisitos funcionales expuestos con anterioridad:

3.1.2.1 Usuario

- Un usuario solo puede ver sus propios usuarios favoritos.
- Un usuario no podrá modificar sus datos de correo, nombre, apellidos y fecha de nacimiento.
- Cualquier usuario puede agregar a cualquier otro como favorito.
- El correo de registro debe ser único en la BD^[A6].
- La contraseña de inicio de sesión debe tener al menos 8 caracteres.
- Un usuario no puede puntuarse a sí mismo.
- Un usuario no puede agregarse de favorito a sí mismo.
- No se mostrará la puntuación de un usuario en caso de que no haya creado ningún viaje.

3.1.2.2 Viajes

- No se podrá crear un viaje con fecha de salida anterior a la fecha actual.
- No se podrá modificar la información acerca de un viaje.
- Cualquier usuario puede ver cualquier viaje creado.
- Si se ha indicado como preferencia de viaje el no aceptar paquetes no se podrán realizar solicitudes de transportar paquetes en ese viaje.
- Sólo se podrán realizar búsquedas de viajes con fecha de salida y llegada posterior a la actual.
- El precio por persona del viaje deberá escogerse dentro de un rango predeterminado por los kilómetros a realizar.
- Un usuario no puede colocarse como pasajero de su propio viaje.

3.1.2.3 Solicitudes

- Un usuario solo puede ver sus propias solicitudes.
- Un usuario no podrá eliminar una solicitud una vez aceptada.

3.1.2.4 Paquetes

- El usuario que ha creado los paquetes es el único que puede ver una lista de estos.
- No se podrá modificar la información de un paquete.
- Cualquiera que conozca el identificador de paquete podrá ver su estado e información.

3.1.2.5 Coches

- El usuario que ha creado los coches es el único que puede ver una lista de estos.
- No se podrá modificar la información de un coche.

3.1.2.6 Notificaciones

- Solo se mostrarán 20 notificaciones en el menú de notificaciones.

3.1.3 Requisitos de rendimiento

El rendimiento de la aplicación debe ser tal que no entorpezca con la acción del usuario, exceptuando los casos de exportado e importado de datos en los que el sistema puede ralentizarse unos segundos hasta la finalización de dicha tarea.

3.2 Modelos de caso de uso

Para describir los distintos comportamientos que tendrá el sistema, usaremos el lenguaje de modelado de sistemas UML^[A8]; que representa los requisitos funcionales del sistema, centrando en que hace y no cómo lo hace.

3.2.1 Diagrama de casos de uso

3.2.1.1 Actores

En nuestro programa solo contamos con un actor: El usuario final. Este es el encargado de llamar a las diferentes funciones del sistema mediante el uso de la interfaz a lo largo de su interacción con el programa.

3.2.1.2 Diagrama

En la siguiente figura se muestra la interacción del usuario con el sistema estructurado como diagrama de caso de uso, en la que el sistema utiliza funciones internas para realizar todas las peticiones del usuario y dando como resultado normalmente, una modificación de la interfaz con la que este interactúa.

En diversas ocasiones la realización de una interacción del usuario tiene como resultado la creación, modificación o eliminación de algún elemento de la base de datos del sistema.

En unos pocos casos, la interacción del usuario tiene como resultado indirecto la creación de una notificación que será almacenada en la base de datos y será mostrada al usuario al cual iba dirigida la acción que se ha realizado.

Como podemos observar en la imagen próxima, es el usuario el que interacciona continuamente con el sistema (líneas continuas) y él, internamente, interactúa consigo mismo (líneas discontinuas) realizando las tareas necesarias para completar la petición del usuario, llegando a realizar la totalidad de las funcionalidades requeridas para el programa.

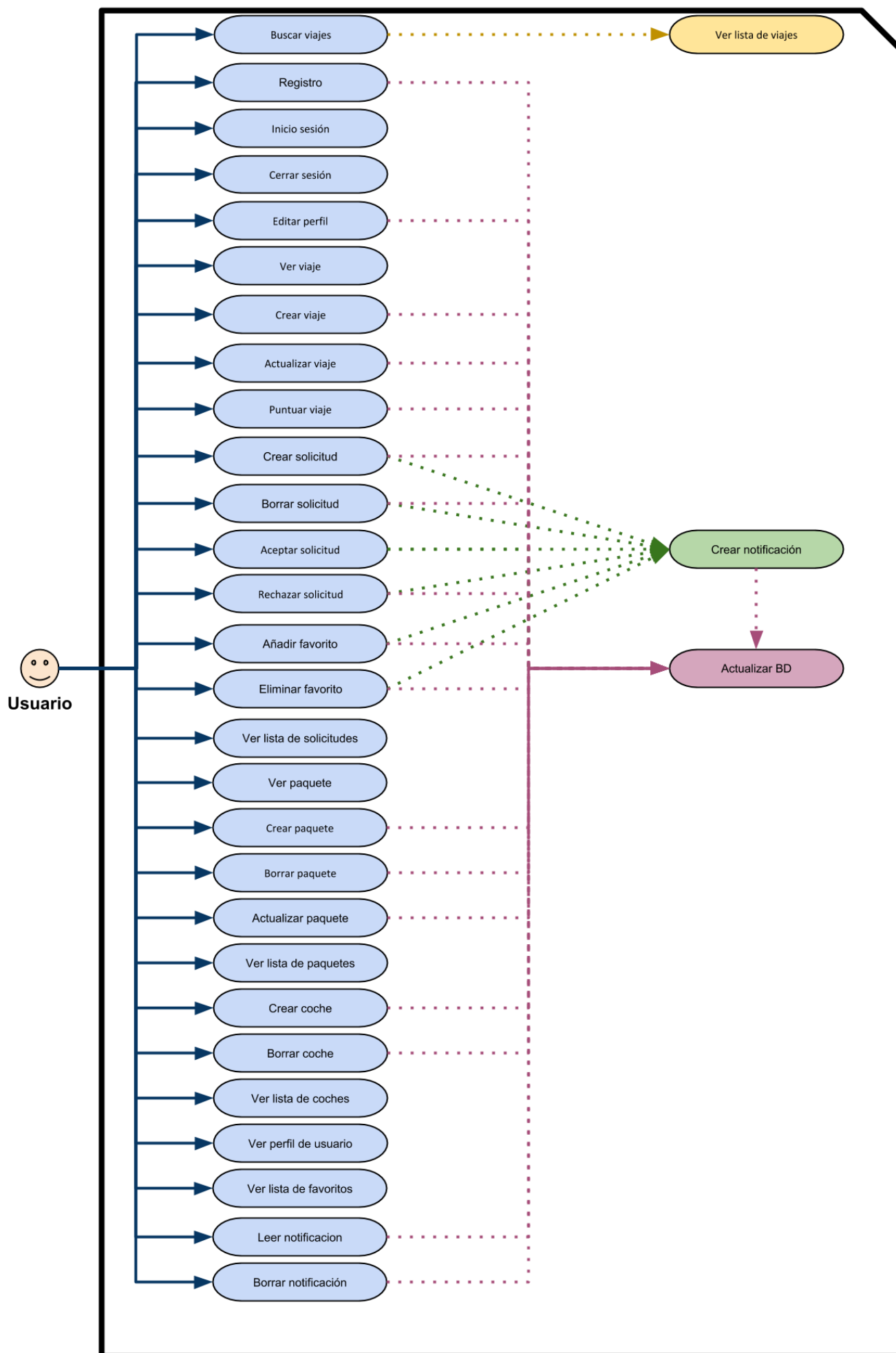


Imagen 3 - Diagrama de caso de uso

3.3 Modelos de comportamiento

Para el correcto funcionamiento de la aplicación, esta debe seguir una serie de pasos al realizar funciones determinadas, llamadas por el usuario. En este apartado veremos con detalle estas funciones y su comportamiento a lo largo de su ejecución.

Los diagramas se han creado haciendo uso de la web *websequencediagrams*^[B22]

- **Registro:** Permite registrar a un nuevo usuario en la aplicación.

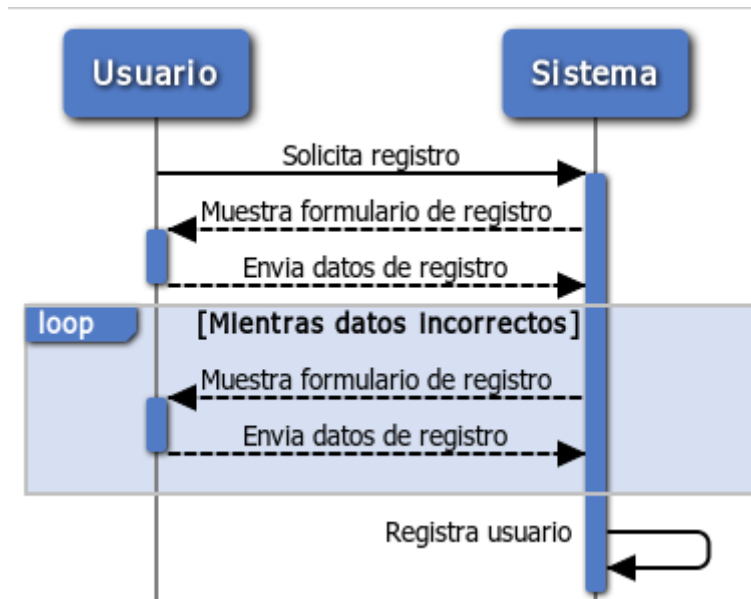


Imagen 4 - Diagrama de secuencia (Registro)

- **Inicio de sesión:** Permite a un usuario iniciar sesión dentro de la aplicación.

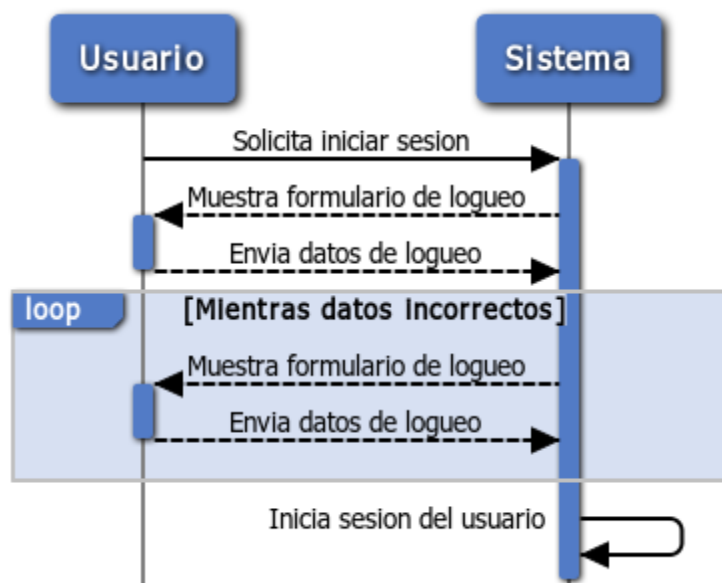


Imagen 5 - Diagrama de secuencia (Inicio de sesión)

- **Cerrar sesión:** Termina la sesión de un usuario dentro de la aplicación.

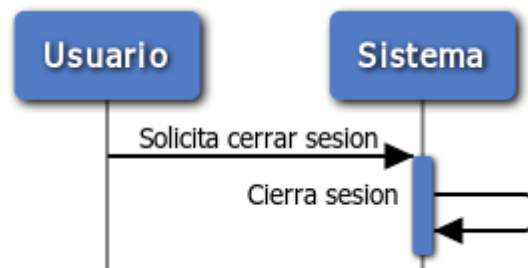


Imagen 6 - Diagrama de secuencia (Cerrar sesión)

- **Editar perfil:** Permite modificar algunos datos del perfil del usuario que tiene iniciada la sesión.

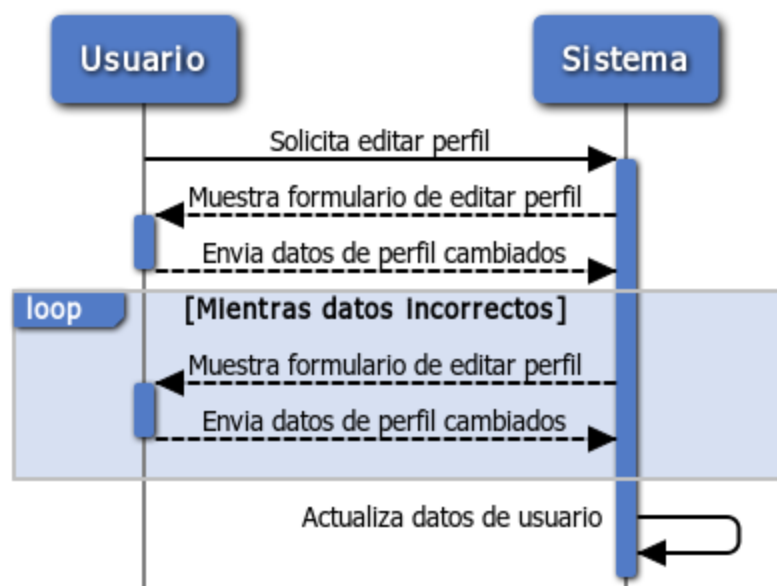


Imagen 7 - Diagrama de secuencia (Editar perfil)

- **Ver viaje:** Muestra información detallada acerca de un viaje.

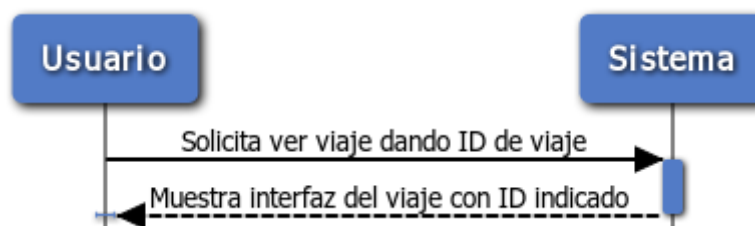


Imagen 8 - Diagrama de secuencia (Ver viaje)

- **Buscar viajes:** Permite registrar a un nuevo usuario en la aplicación.

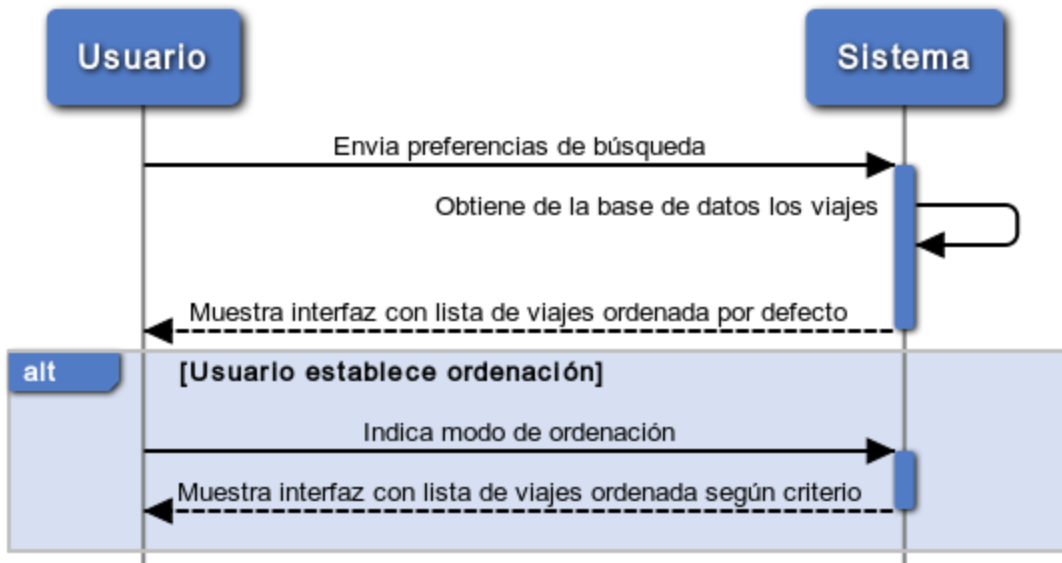


Imagen 9 - Diagrama de secuencia (Buscar viaje)

- **Crear viaje:** Permite registrar a un nuevo usuario en la aplicación.

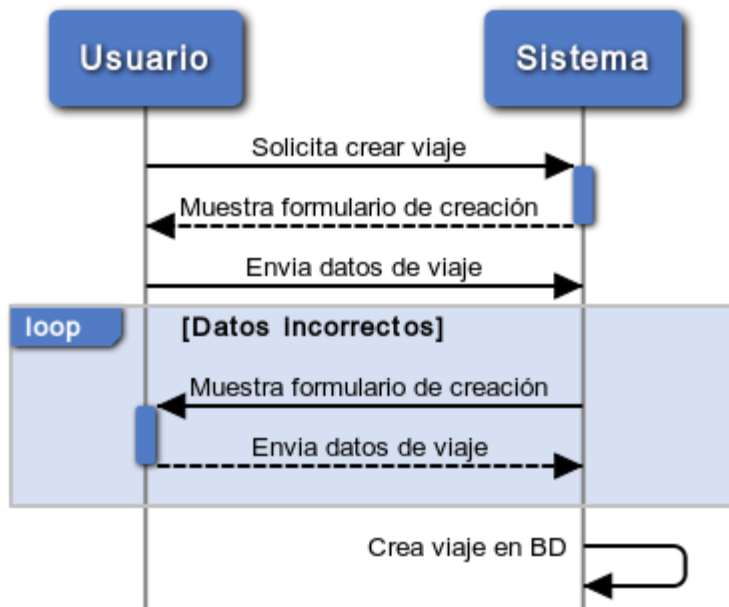


Imagen 10 - Diagrama de secuencia (Crear viaje)

- **Actualizar viaje:** Permite Un usuario puede cambiar algún parámetro del viaje que ha creado.

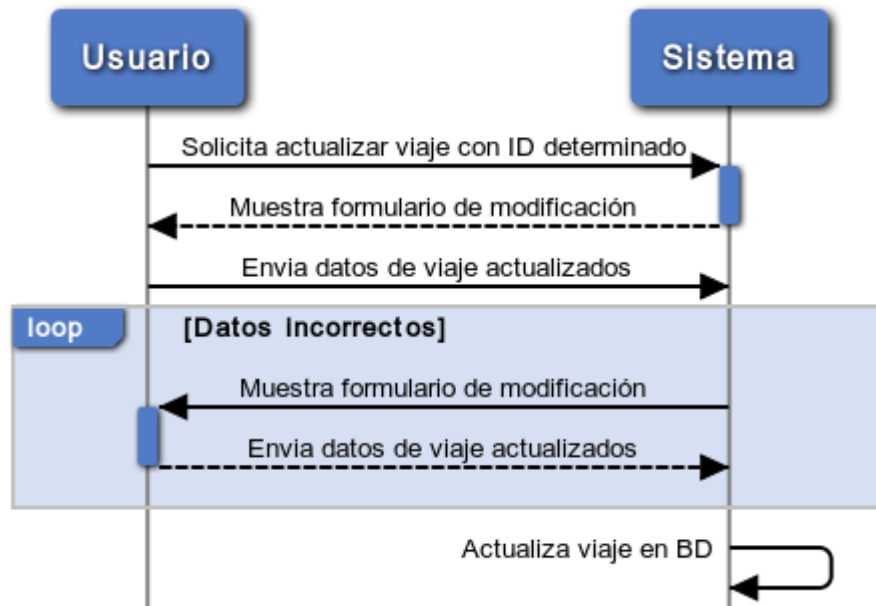


Imagen 11 - Diagrama de secuencia (Actualizar viaje)

- **Puntuar viaje ajeno:** Permite a un usuario puntuar un viaje del cual ha sido pasajero.

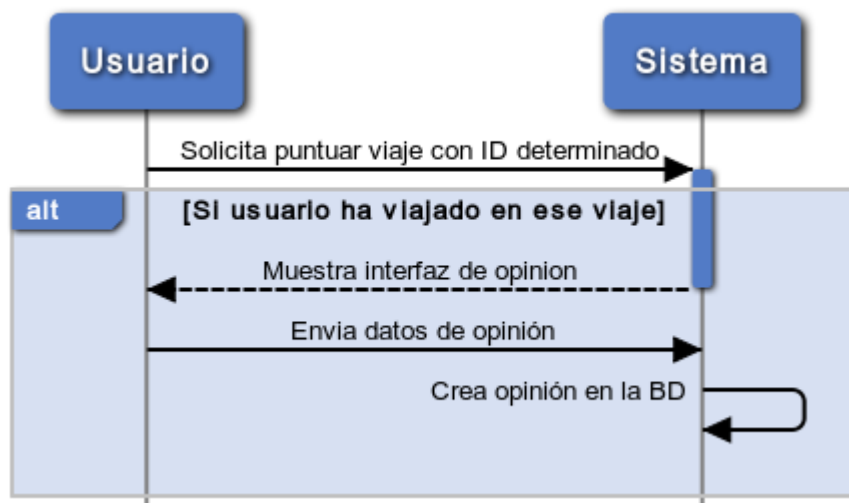


Imagen 12 - Diagrama de secuencia (Puntuar viaje)

- **Ver viajes de un usuario:** Muestra una lista de viajes pertenecientes a un usuario.

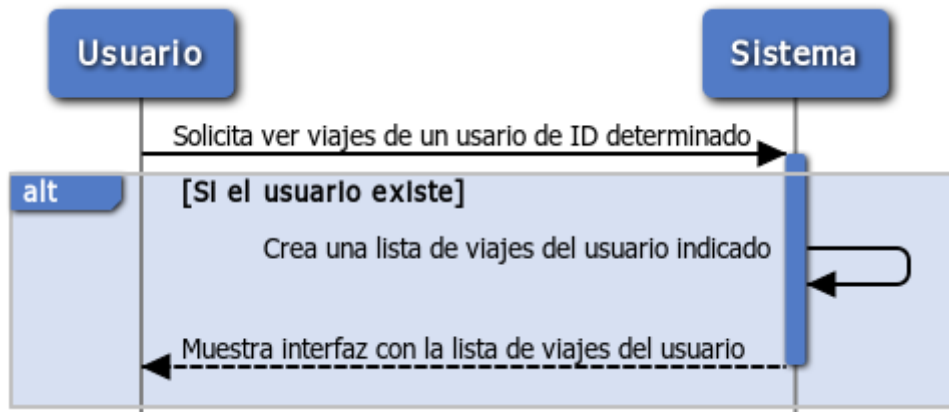


Imagen 13 - Diagrama de secuencia (Ver lista de viajes de un usuario)

- **Crear solicitud:** Crea una nueva solicitud para un viaje.

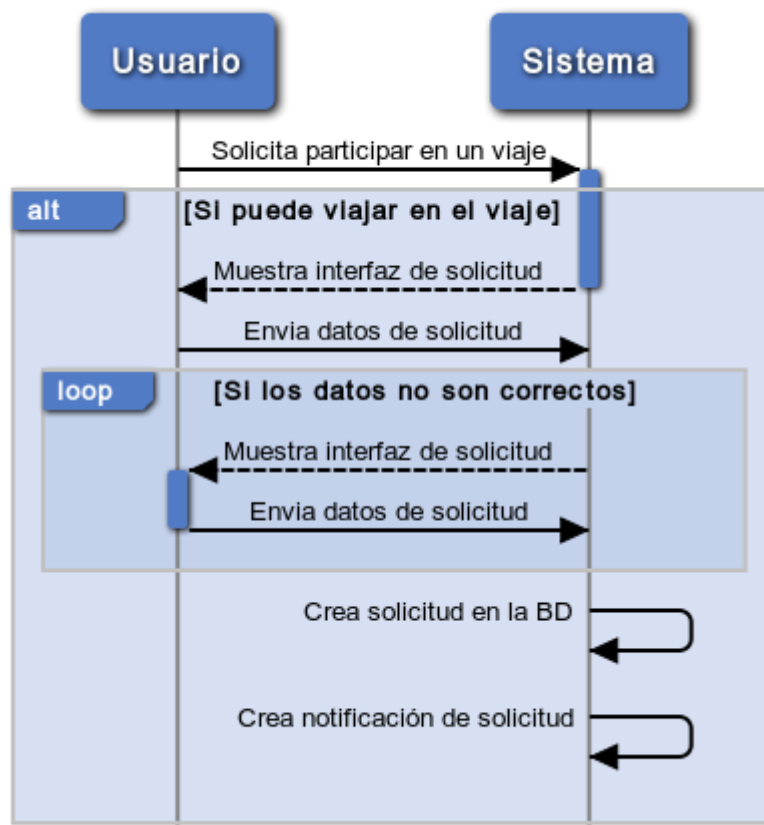


Imagen 14 - Diagrama de secuencia (Crear solicitud)

- **Borrar solicitud propia:** Elimina una solicitud que ha creado el mismo usuario anteriormente.

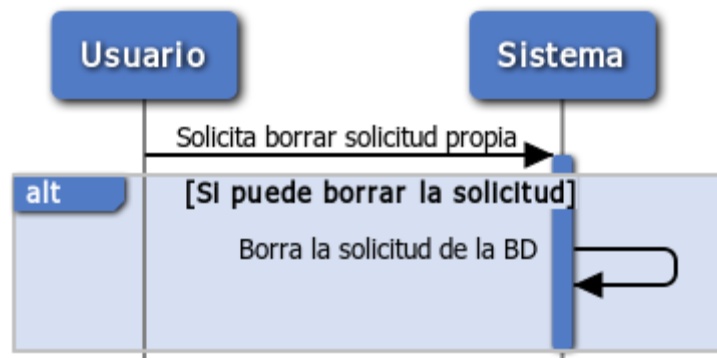


Imagen 15 - Diagrama de secuencia (Borrar solicitud)

- **Aceptar solicitud:** Acepta una solicitud de un usuario para viajar en el viaje que ha creado el usuario que la acepta.

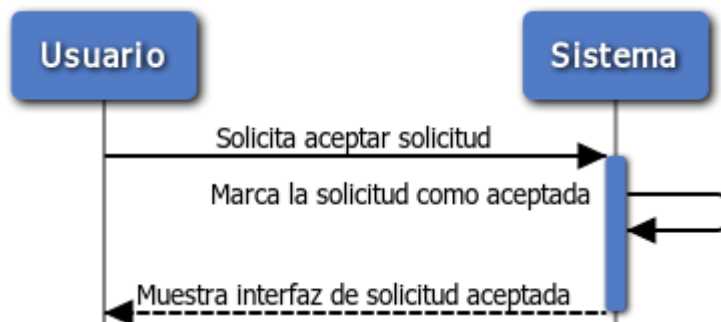


Imagen 16 - Diagrama de secuencia (Aceptar solicitud)

- **Rechazar solicitud:** Rechaza una solicitud de un usuario para viajar en el viaje que ha creado el usuario que la rechaza.

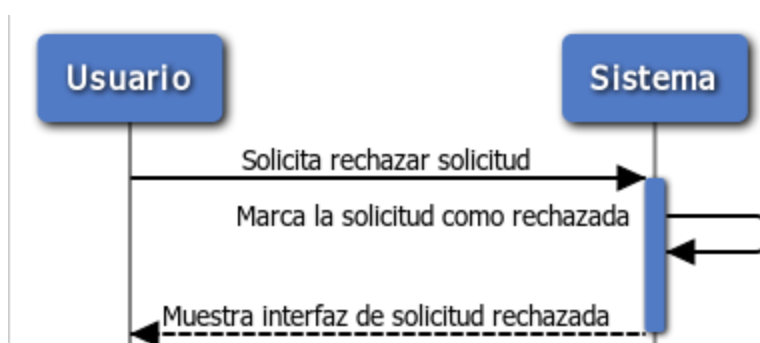


Imagen 17 - Diagrama de secuencia (Rechazar solicitud)

- **Ver lista de solicitudes:** Muestra una lista de solicitudes que tiene un usuario.

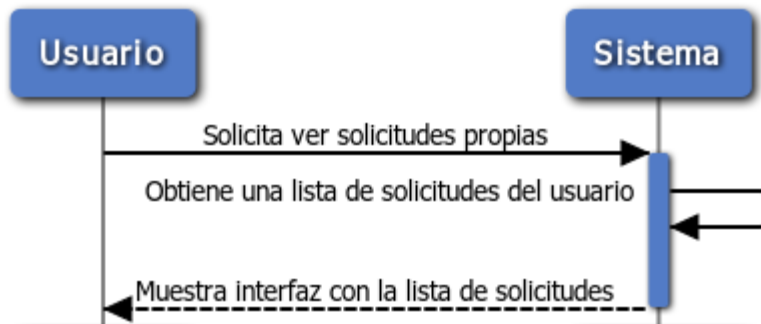


Imagen 18 - Diagrama de secuencia (Lista de solicitudes)

- **Ver paquete:** Muestra información detallada acerca de un paquete.

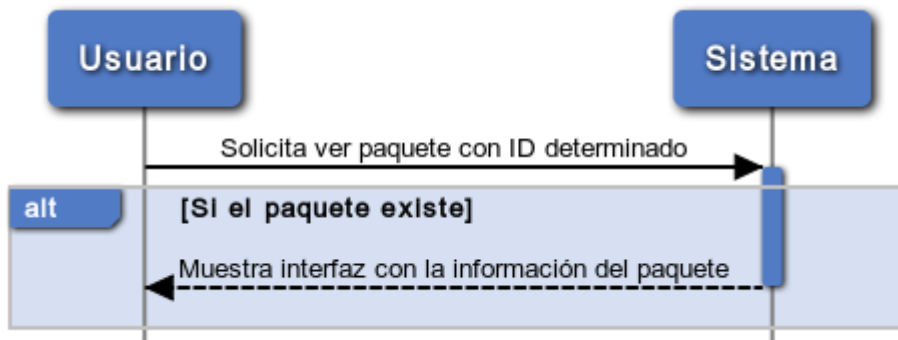


Imagen 19 - Diagrama de secuencia (Ver paquete)

- **Crear paquete:** Crea un nuevo paquete con los datos indicados.

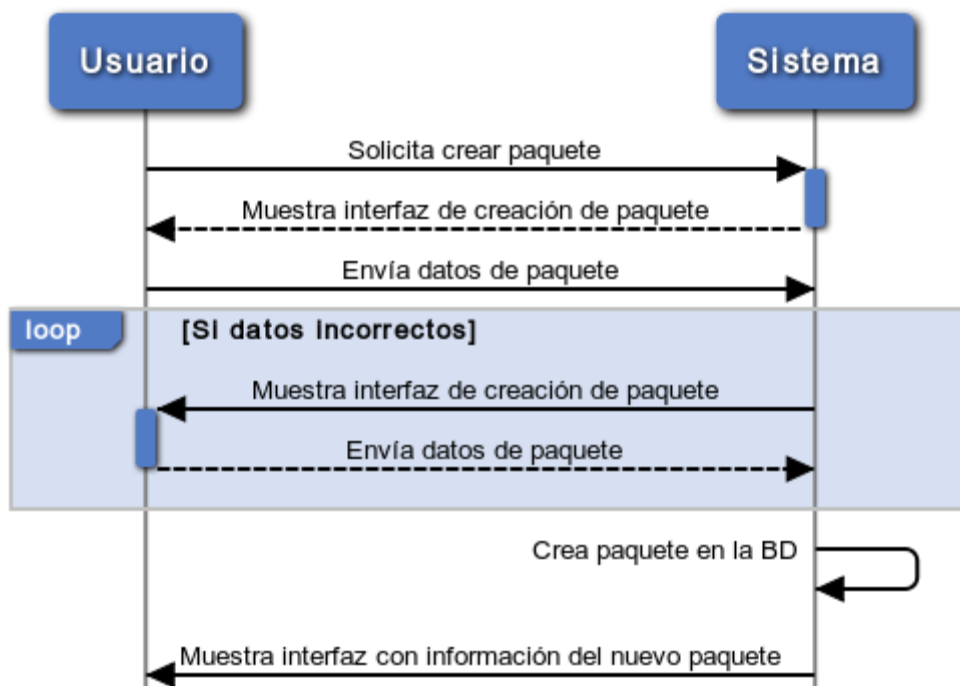


Imagen 20 - Diagrama de secuencia (Crear paquete)

- **Borrar paquete:** Elimina un paquete.

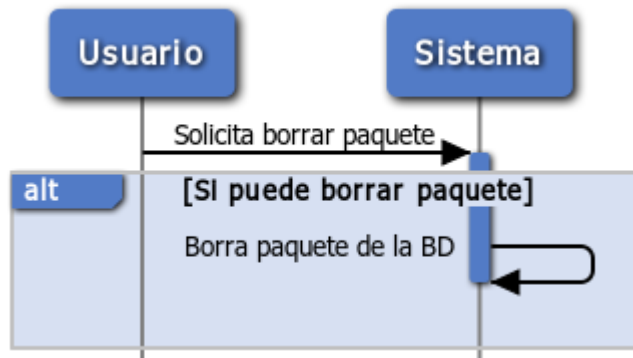


Imagen 21 - Diagrama de secuencia (Borrar paquete)

- **Actualizar estado de paquete:** Actualiza la información de estado de un paquete.

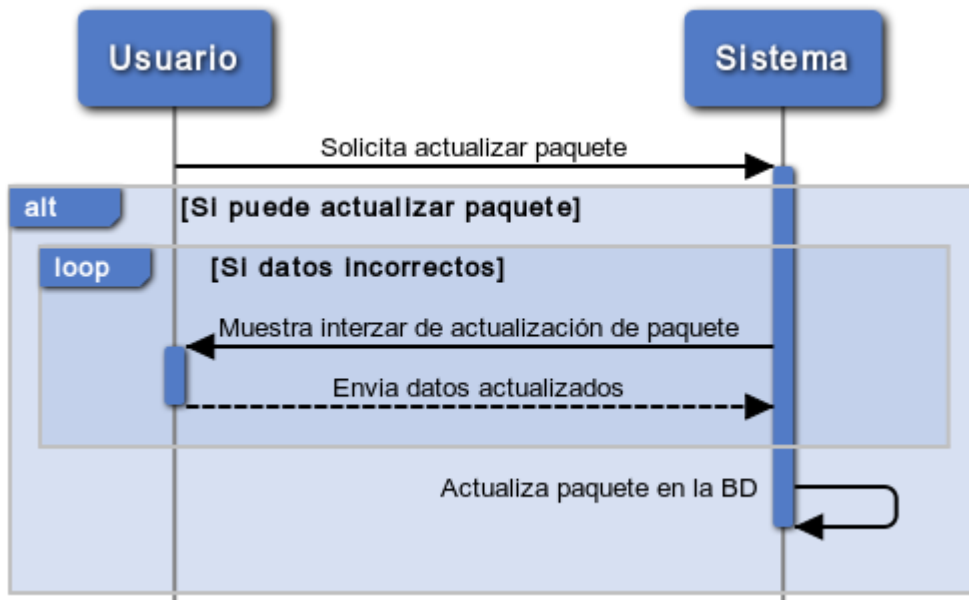


Imagen 22 - Diagrama de secuencia (Actualizar paquete)

- **Ver lista de paquetes:** Muestra una lista de paquetes que ha creado un usuario.

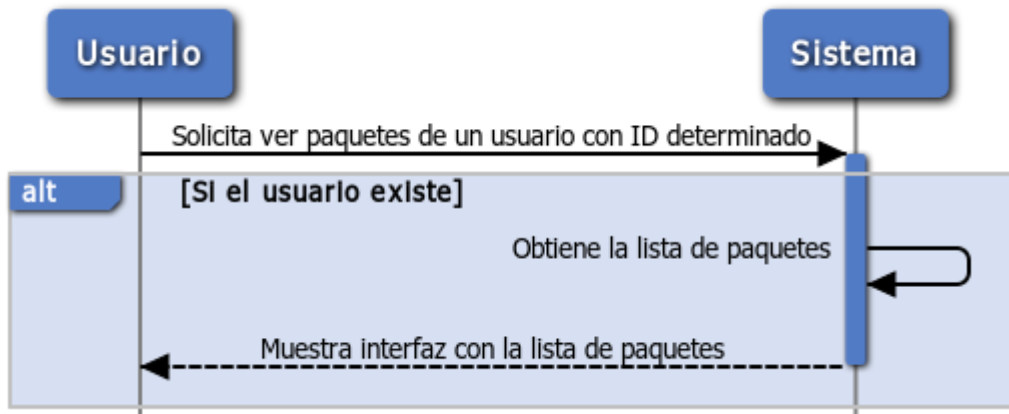


Imagen 23 - Diagrama de secuencia (Lista de paquetes)

- **Crear coche:** Crea un nuevo coche asociado a un usuario.

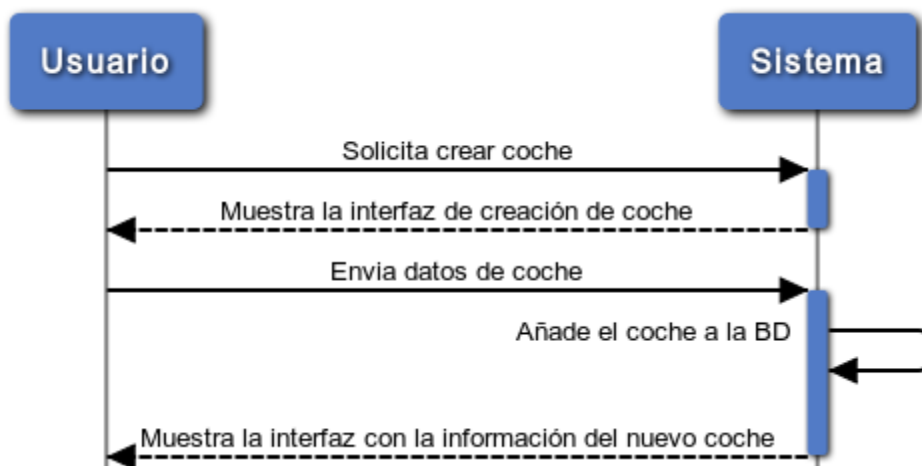


Imagen 24 - Diagrama de secuencia (Crear coche)

- **Borrar coche:** Elimina al usuario como conductor de un coche.

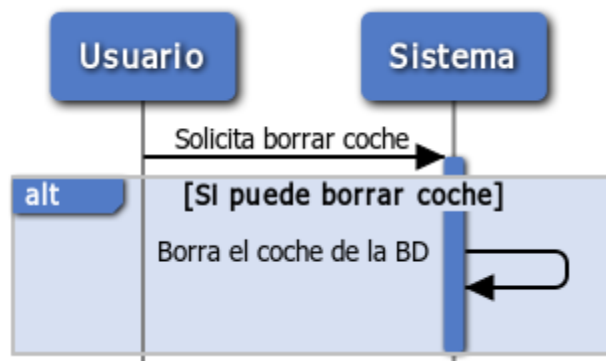


Imagen 25 - Diagrama de secuencia (Borrar coche)

- **Ver lista de coches:** Muestra una lista de coches de los que el usuario es conductor.

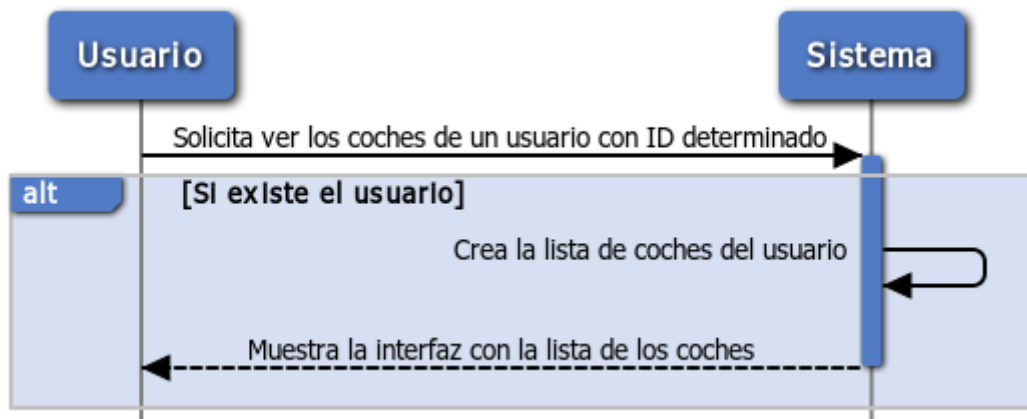


Imagen 26 - Diagrama de secuencia (Lista de coches)

- **Ver perfil de usuario:** Muestra la información de un usuario.

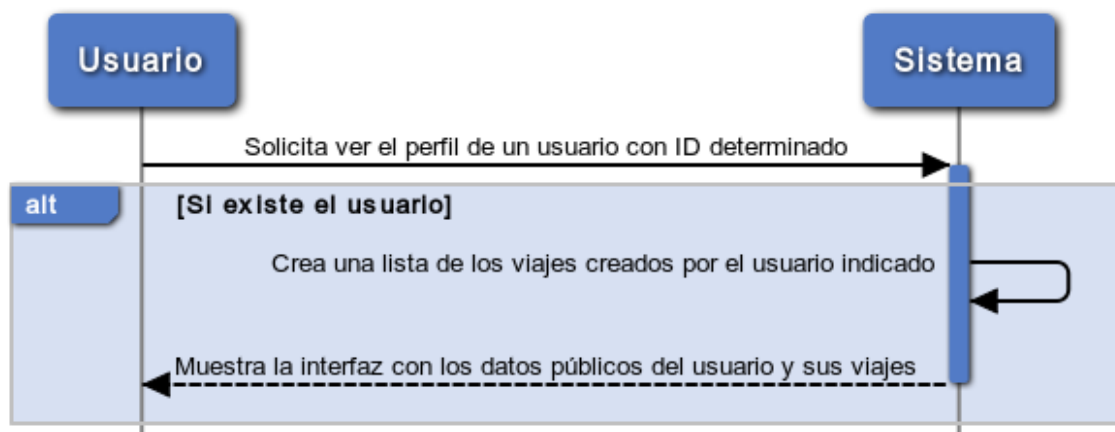


Imagen 27 - Diagrama de secuencia (Perfil de usuario)

- **Ver lista de usuario favoritos:** Muestra la lista de usuarios agregados como favoritos por otro usuario.

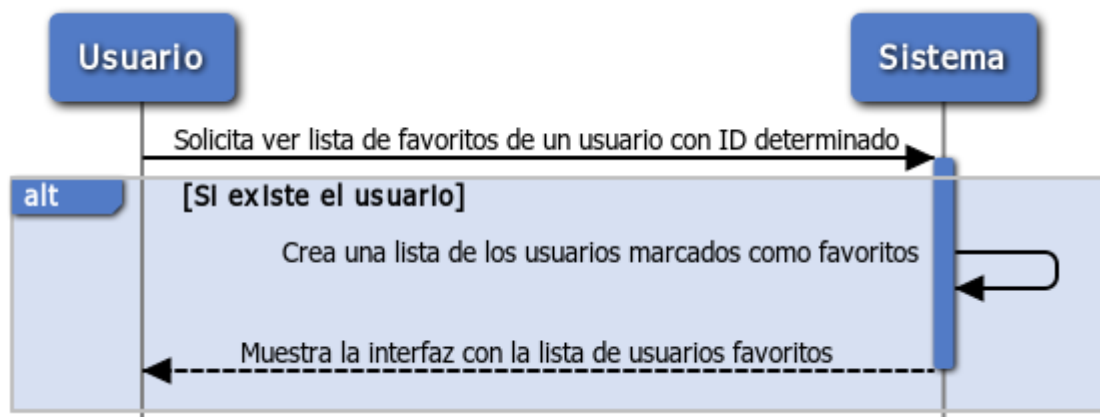


Imagen 28 - Diagrama de secuencia (Lista de usuarios favoritos)

- **Añadir favorito:** Añade como favorito a un usuario.

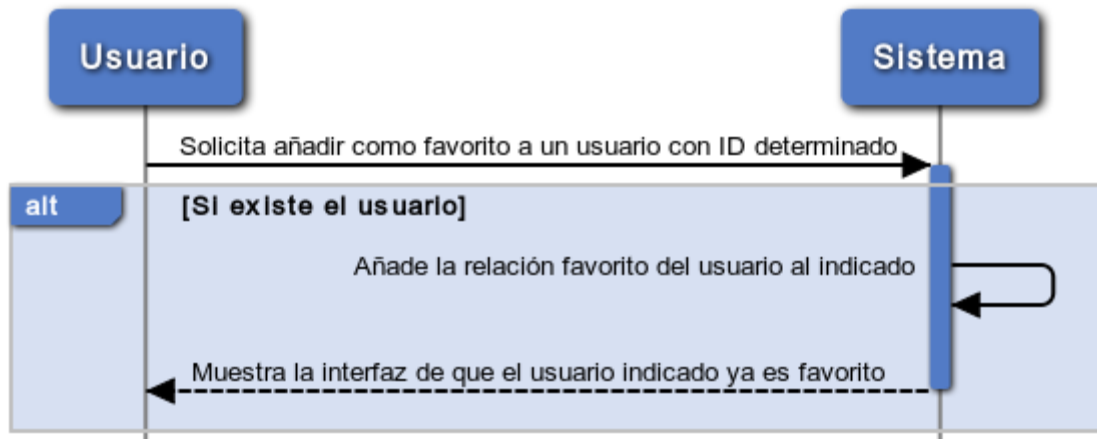


Imagen 29 - Diagrama de secuencia (Añadir favorito)

- **Eliminar favorito:** Elimina de favoritos a un usuario.

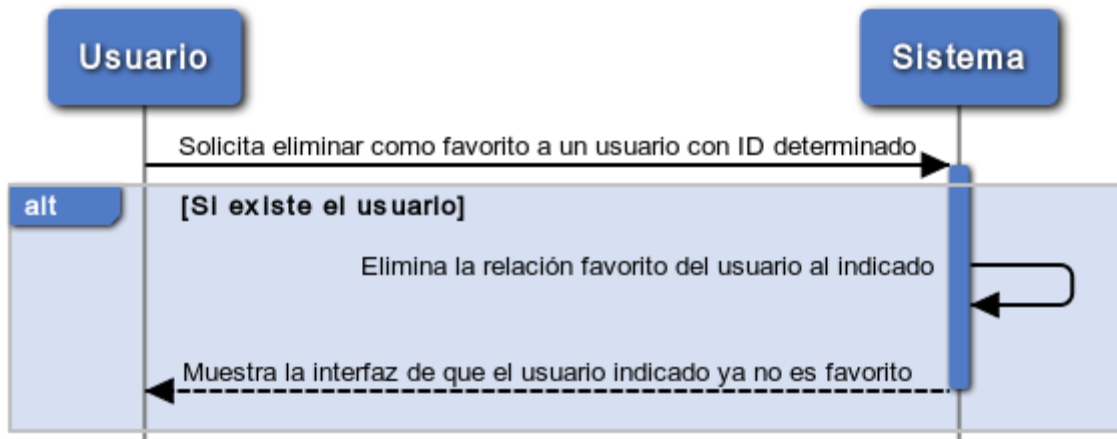


Imagen 30 - Diagrama de secuencia (Eliminar favorito)

- **Leer notificación:** Marca una notificación como leída.

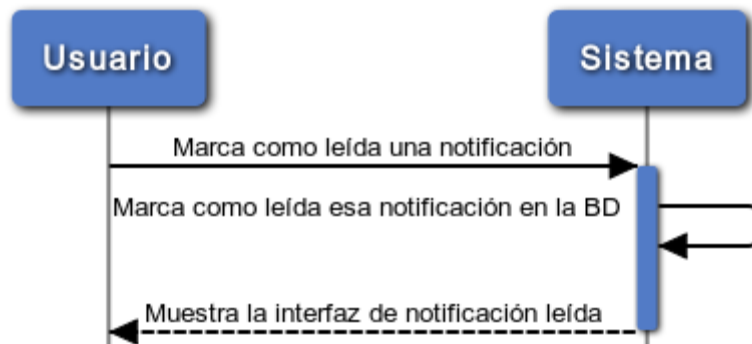


Imagen 31 - Diagrama de secuencia (Leer notificación)

- **Borrar notificación:** Elimina una notificación.

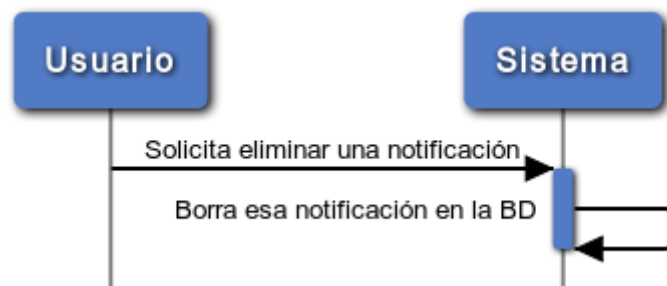


Imagen 32 - Diagrama de secuencia (Borrar notificación)

4. Diseño

4.1 Diseño de Base de Datos

Para el correcto funcionamiento de la aplicación se ha hecho uso de una base de datos para almacenar toda la información.

4.1.1 Diagrama Entidad-Relación inicial

En primer lugar se creó un diagrama general de entidad-relación para tener una idea general acerca de las tablas a crear en la BD^[A6]. Esto se puede ver en la imagen siguiente.

Cuenta con los siguientes elementos:

- Los cuadrados azules identifican las entidades.
- Los cuadrados azules de doble borde identifican las entidades débiles.
- Los rombos indican relaciones entre entidades.
- Los recuadros de colores que engloban entidades junto con sus relaciones indican agregaciones.
- Cada entidad, relación o agregación puede contar con atributos, que se marcan con un punto unido a esta y su nombre.
- Cada relación muestra su cardinalidad en el extremo que la une con otro elemento del diagrama.

La mayoría de identificadores no se muestran en el esquema pero sí en las tablas.

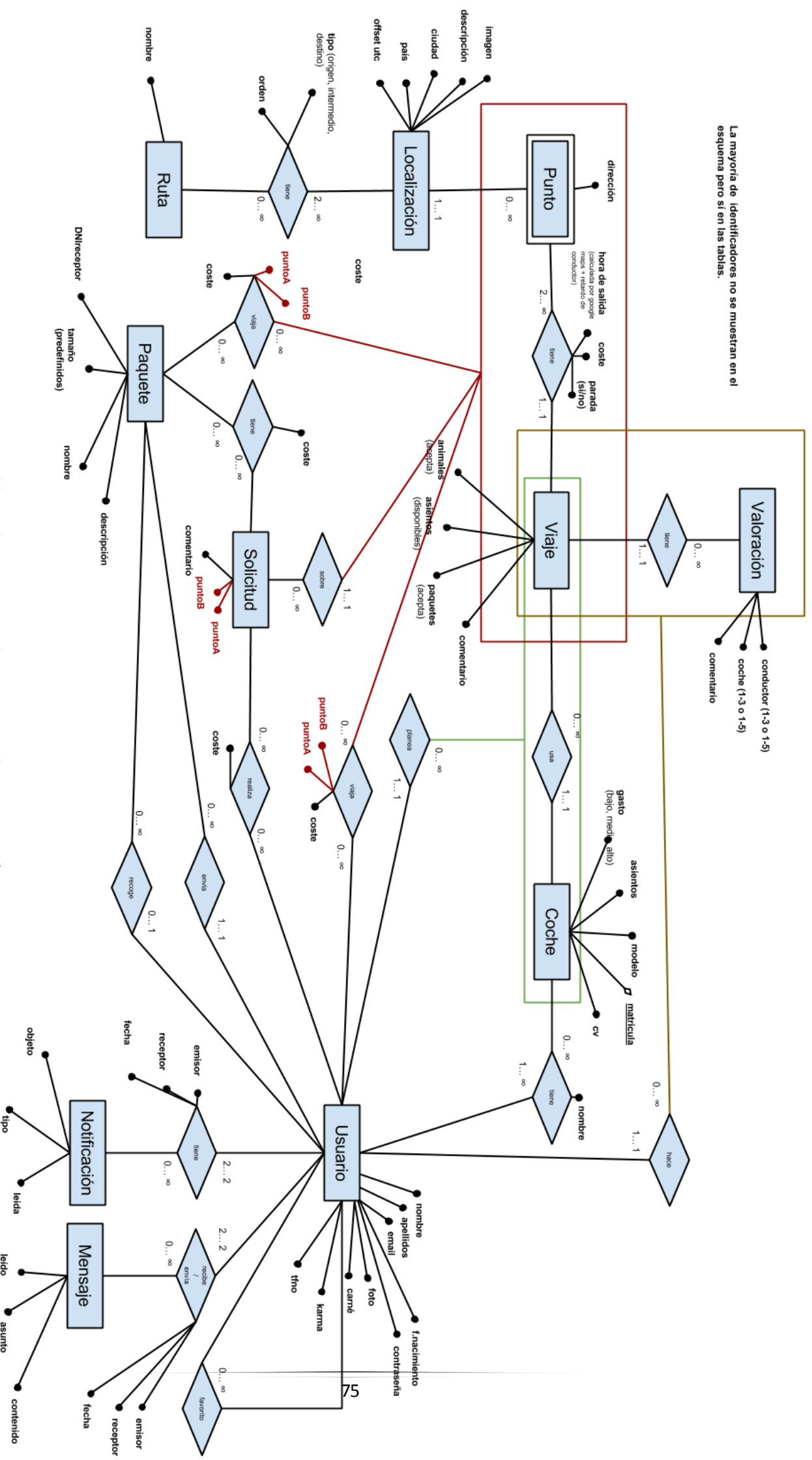


Imagen 33 - Diseño BD – Diagrama Entidad-Relación

4.1.2 Diagrama de Entidad-Relación Mejorado

Una vez creado un diagrama general, se crearon las tablas pertinentes en la *BD*^[A6]. La siguiente imagen muestra el diagrama usado finalmente en la *BD*^[A6], en formato *UML*^[A8].

En esta imagen se muestran:

- Las tablas de la *BD*^[A6] con sus atributos, cada uno con los siguientes datos:
 - Tipo de atributo:
 - Según color:
 - **Azul**: Atributo normal.
 - **Rojo**: Pertenece a una clave extranjera.
 - Según forma:
 - **Llave**: Clave primaria
 - **Rombo con fondo blanco**: Atributo que puede ser *null*.
 - **Rombo relleno**: Atributo que no puede ser *null*.
 - Nombre
 - Tipo de dato y su longitud
- Relaciones entre tablas que pueden ser:
 - **Líneas continuas**: El atributo forma parte de la clave primaria.
 - **Líneas discontinuas**: El atributo no forma parte de la clave primaria.

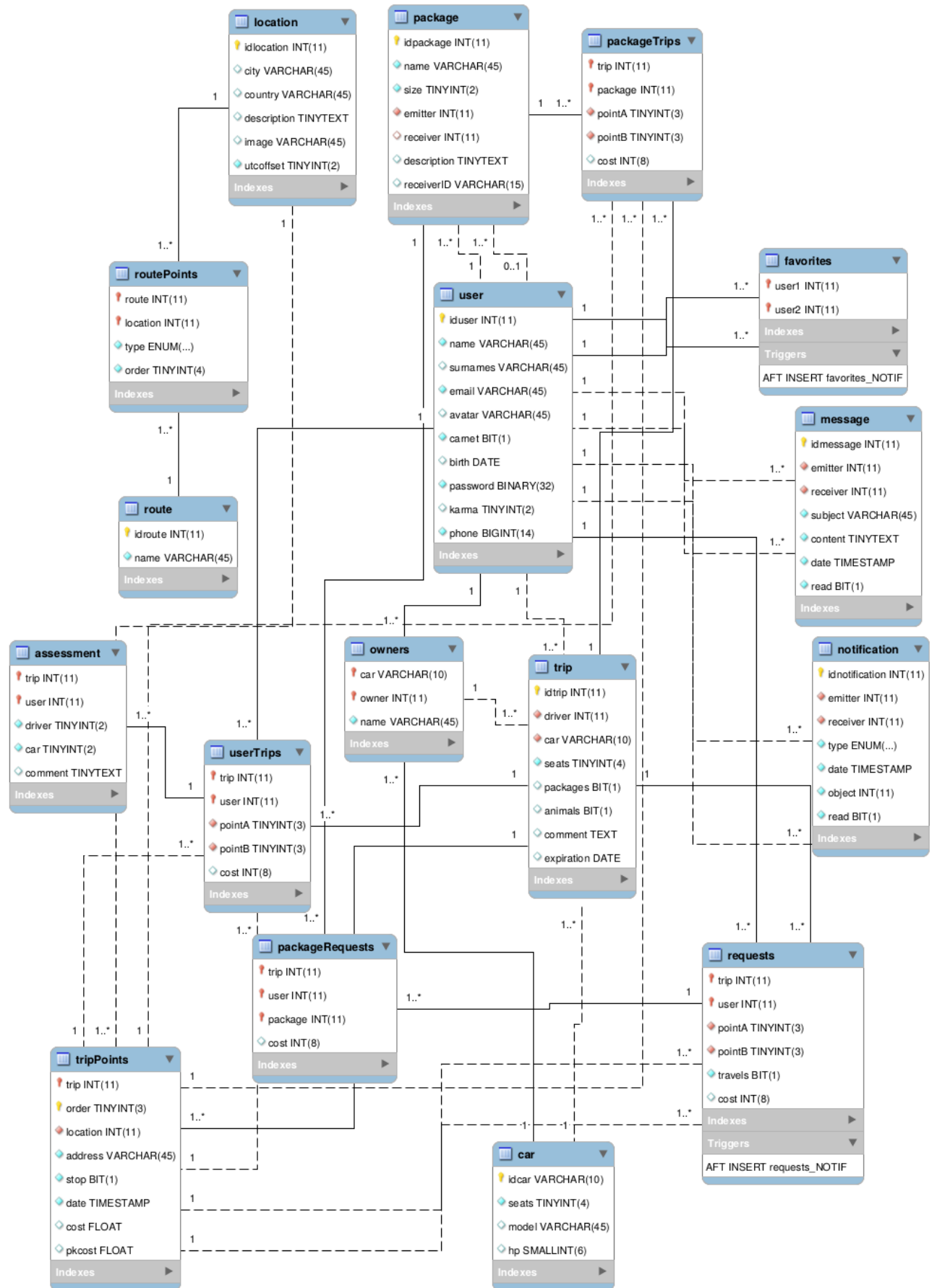


Imagen 34 - Diseño BD – Diagrama Entidad-Relación mejorado

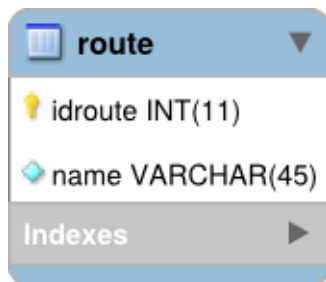
4.1.3 Tablas

A continuación explicaremos en detalle cada una de las tablas utilizadas en la BD^[A6]. En primer lugar aquellas pertenecientes a las entidades y, en segundo lugar aquellas pertenecientes a relaciones.

4.1.3.1 route

Esta tabla representa las rutas existentes en la aplicación.

4.1.3.1.1 Columnas



- **idroute**: Identificador único y autoincremental de ruta.
- **name**: Nombre de la ruta.

Imagen 35 - Tabla BD - route

4.1.3.2 location

Esta tabla representa una localización, que identifica una ciudad.

4.1.3.2.1 Columnas

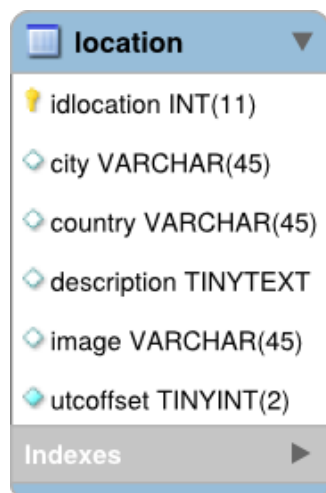


Imagen 36 - Tabla BD - location

- **idlocation**: Identificador único y autoincremental de localización.
- **city**: Nombre de la ciudad.
- **country**: País en el que se encuentra la ciudad.
- **description**: Texto que guarda una pequeña descripción turística de la ciudad.
- **image**: Ruta en la que se encuentra la imagen que representa la ciudad.
- **utcoffset**: Desfase horario de la ciudad con respecto al UTC^[A9] 0.

4.1.3.3 trip

Esta tabla almacena los viajes.

4.1.3.3.1 Columnas

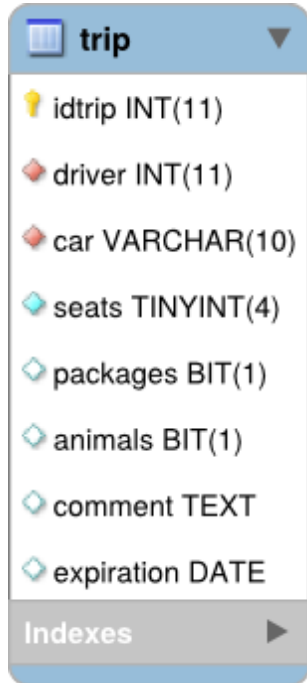


Imagen 37 - Tabla BD - trip

- **idtrip**: Identificador único y autoincremental de viaje.
- **driver**: Identificador único que representa al usuario que ha creado el viaje y será el conductor.
- **car**: Identificador único del coche que se usará en el viaje.
- **seats**: Número de asientos disponibles para su uso por otros usuarios.
- **packages**: Indicador de si se permite transportar paquetes o no en el viaje.
- **animals**: Indicador de si se permite transportar animales o no en el viaje.
- **comment**: Texto comentando algún detalle del viaje creado por el conductor.
- **expiration**: Indica la fecha de expiración del viaje.

4.1.3.3.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_trip_1</i>	user	driver	iduser
<i>fk_trip_2</i>	car	car	car
<i>fk_trips_3</i>	owners	driver	owner
		car	car

4.1.3.4 user

Esta tabla representa a un usuario.

4.1.3.4.1 Columnas

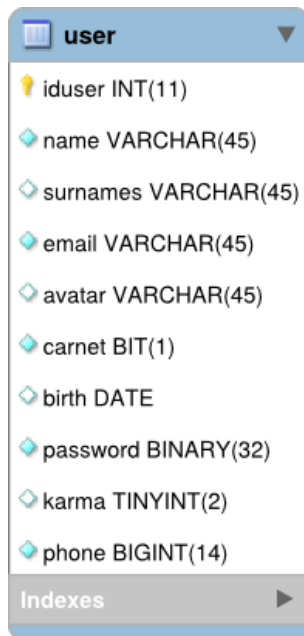


Imagen 38 - Tabla BD - user

- **iduser**: Identificador único y autoincremental de usuario.
- **name**: Nombre del usuario.
- **surnames**: Apellidos del usuario.
- **email**: Correo electrónico del usuario.
- **avatar**: Ruta en la que se encuentra la imagen de perfil del usuario.
- **carnet**: Si posee o no carnet de conducir.
- **password**: Contraseña del usuario.
- **karma**: Media de puntuaciones otorgadas al usuario.
- **phone**: Número de teléfono del usuario.

4.1.3.5 car

Esta tabla representa a un coche que será usado por uno o varios usuarios en viajes.

4.1.3.5.1 Columnas

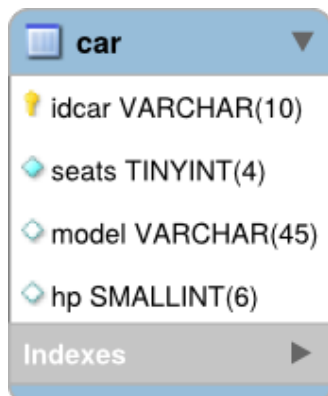


Imagen 39 - Tabla BD - car

- **idcar**: Identificador único y autoincremental de coche.
- **seats**: Asientos que tiene el coche (Incluido el del conductor).
- **model**: Modelo del coche.
- **hp**: Potencia en caballos del coche.

4.1.3.6 package

Esta tabla contendrá los paquetes creados por los usuarios.

4.1.3.6.1 Columnas

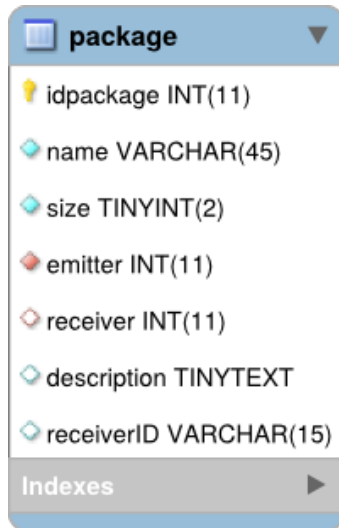


Imagen 40 - Tabla BD - package

- **idpackage**: Identificador único y autoincremental de paquete.
- **name**: Nombre del paquete.
- **size**: Número indicador del tamaño aproximado del paquete.
- **emitter**: Identificador único de usuario que envía el paquete.
- **receiver**: Identificador único de usuario que recibirá el paquete al final del viaje.
- **description**: Texto comentando detalles acerca del paquete.
- **receiverID**: Identificador del receptor del paquete (DNI para españoles, por ejemplo)

4.1.3.6.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_package_1</i>	user	emitter	iduser
<i>fk_package_2</i>	user	receiver	iduser

4.1.3.7 notification

Esta tabla contiene todas las notificaciones generadas por el sistema.

4.1.3.7.1 Columnas

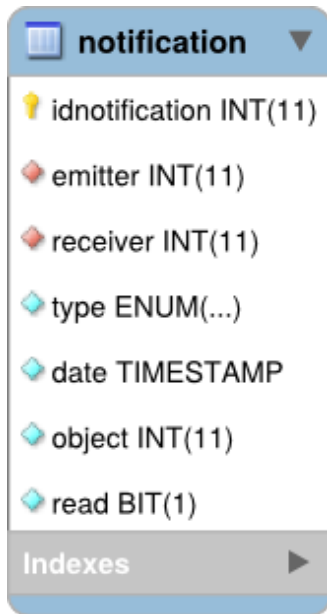


Imagen 41 - Tabla BD - notification

- **idnotification**: Identificador único y autoincremental de notificación.
- **emitter**: Identificador único de usuario que ha sido responsable de la creación de la notificación.
- **receiver**: Identificador único del usuario que recibe la notificación.
- **type**: Enumerado que indica el tipo de notificación.
- **date**: Fecha de la notificación, establecida, automáticamente, cada vez que ésta es modificada con el valor de la fecha en ese momento del servidor.
- **object**: Identificador del objeto representado por la notificación (solicitud, viaje, etc.).
- **read**: Indica si la notificación ha sido leída o no.

4.1.3.7.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_notification_1</i>	user	emitter	iduser
<i>fk_notification_2</i>	user	receiver	iduser

4.1.3.8 requests

Esta tabla contiene las solicitudes realizadas por los usuarios.

4.1.3.8.1 Columnas

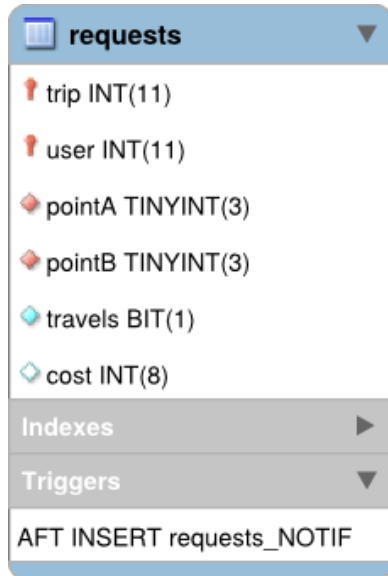


Imagen 42 - Tabla BD - requests

- **trip**: Identificador del viaje al cual se hace la solicitud.
- **user**: Identificador del usuario que crea la solicitud.
- **pointA**: Identificador del punto de salida, dentro del viaje, del que quiere salir el usuario.
- **pointB**: Identificador del punto de llegada, dentro del viaje, en el que quiere bajarse el usuario.
- **travels**: Si el usuario viaja o no como pasajero.
- **cost**: Coste pactado por el viaje como pasajero.

4.1.3.8.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_requests_1</i>	trip	trip	idtrip
<i>fk_requests_2</i>	user	user	iduser
<i>fk_requests_3</i>	tripPoints	trip	trip
		pointA	order
<i>fk_requests_4</i>	tripPoints	trip	trip
		pointB	order

4.1.3.8.3 Triggers

Al insertar una nueva fila en la tabla se lanza un evento de creación de una notificación con los datos:

- **emitter**: user
- **receiver**: trip.driver
- **type**: 'request'
- **date**: Establecida automáticamente.

4.1.3.9 assessment

Esta tabla representa una valoración hecha por un usuario acerca de un viaje.

4.1.3.9.1 Columnas



- **trip**: Identificador del viaje sobre el que se hace la valoración.
- **user**: Identificador del usuario que realiza la valoración.
- **driver**: Valoración numérica sobre el conductor del viaje.
- **car**: Valoración numérica sobre el coche usado en el viaje.
- **comment**: Texto comentando la valoración.

Imagen 43 - Tabla BD - assessment

4.1.3.9.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_assessment_1</i>	userTrips	trip	trip
		user	user

4.1.3.10 message

Esta tabla contiene los mensajes mandados de un usuario a otro en la aplicación.

4.1.3.10.1 Columnas

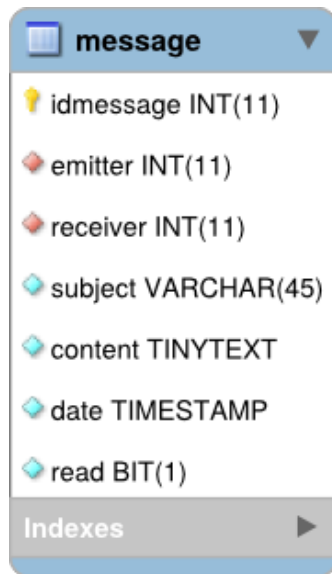


Imagen 44 - Tabla BD - message

- **idmessage**: Identificador único y autoincremental de mensaje.
- **emitter**: Identificador del usuario que envía el mensaje.
- **receiver**: Identificador del usuario que recibe el mensaje.
- **subject**: Título del mensaje.
- **content**: Texto con el contenido del mensaje.
- **date**: Fecha del mensaje, establecida, automáticamente, cada vez que ésta es modificada con el valor de la fecha en ese momento del servidor
- **read**: Indicador de si el mensaje ha sido leído o no.

4.1.3.10.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_message_1</i>	user	emitter	iduser
<i>fk_message_2</i>	user	receiver	iduser

4.1.3.10.3 Triggers

Al insertar una nueva fila en la tabla se lanza un evento de creación de una notificación con los datos:

- **emitter**: emitter
- **receiver**: receiver
- **type**: 'message'
- **date**: Establecida automáticamente.

4.1.3.11 owners

Esta tabla contiene la relación entre un usuario y el coche del cual es propietario.

4.1.3.11.1 Columnas

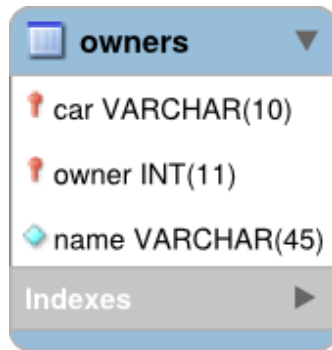


Imagen 45 - Tabla BD - owners

- **idmessage**: Identificador único y autoincremental de mensaje.
- **emitter**: Identificador del usuario que envía el mensaje.
- **receiver**: Identificador del usuario que recibe el mensaje.
- **subject**: Título del mensaje.
- **content**: Texto con el contenido del mensaje.
- **date**: Fecha de creación del mensaje.
- **read**: Indicador de si el mensaje ha sido leído o no.

4.1.3.11.2 Claves extranjeras

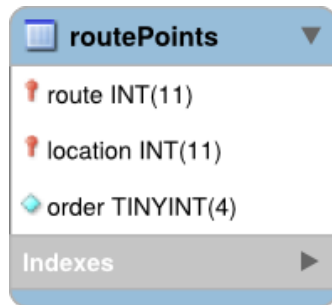
Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_owners_1</i>	car	car	car
<i>fk_owners_2</i>	user	owner	iduser

4.1.3.12 routePoints

Esta tabla contiene los puntos de cada viaje con su orden y localización a la que corresponden.

4.1.3.12.1 Columnas



- **route**: Identificador único de la ruta al que pertenece el punto.
- **location**: Identificador único de la localización a la que pertenece el punto.
- **order**: Número de ordenación del punto dentro de la ruta.

Imagen 46 - Tabla BD - routePoints

4.1.3.12.2 Claves extranjeras

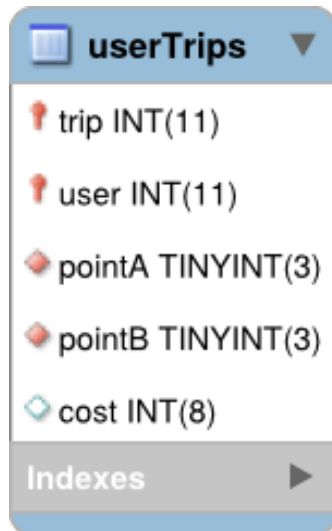
Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_routePoints_1</i>	route	route	idroute
<i>fk_routePoints_2</i>	location	location	idlocation

4.1.3.13 userTrips

Esta tabla contiene los viajes realizados o a realizar por un usuario desde un punto de salida hacia uno de destino.

4.1.3.13.1 Columnas



- **trip**: Identificador del viaje en el cual se va a viajar.
- **user**: Identificador del usuario que va a viajar como pasajero.
- **pointA**: Identificador del punto de salida.
- **pointB**: Identificador del punto de destino.
- **cost**: Coste del viaje.

Imagen 47 - Tabla BD - userTrips

4.1.3.13.2 Claves extranjeras

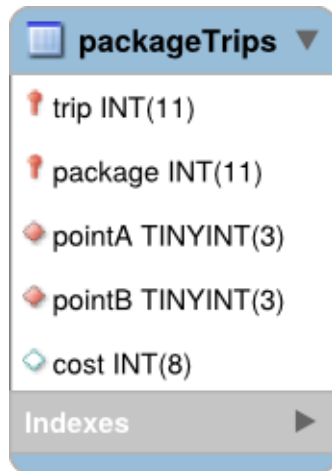
Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_userTrips_1</i>	trip	trip	idtrip
<i>fk_userTrips_2</i>	user	user	iduser
<i>fk_userTrips_3</i>	tripPoints	trip	trip
		pointA	order
<i>fk_userTrips_4</i>	tripPoints	trip	trip
		pointB	order

4.1.3.14 packageTrips

Esta tabla contiene los paquetes que han sido o van a ser enviados en un viaje, desde un punto de salida a uno de destino.

4.1.3.14.1 Columnas



- **trip**: Identificador del viaje en el cual se va a viajar.
- **package**: Identificador del paquete que va a viajar como equipaje.
- **pointA**: Identificador del punto de salida.
- **pointB**: Identificador del punto de destino.
- **cost**: Coste del viaje.

Imagen 48 - Tabla BD - packageTrips

4.1.3.14.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_packageTrips_1</i>	trip	trip	idtrip
<i>fk_packageTrips_2</i>	package	package	idpackage
<i>fk_packageTrips_3</i>	tripPoints	trip	trip
		pointA	order
<i>fk_packageTrips_4</i>	tripPoints	trip	trip
		pointB	order

4.1.3.15 favorites

Esta tabla contiene las relaciones de favorito que ha marcado un usuario hacia otro. Esta relación no tiene por qué ser recíproca.

4.1.3.15.1 Columnas



- **user1**: Identificador del usuario que ha marcado como favorito a otro.
- **user2**: Identificador del usuario que ha sido marcado como favorito.

Imagen 49 - Tabla BD - favorites

4.1.3.15.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_favorites_1</i>	user	user1	iduser
<i>fk_favorites_2</i>	user	user2	iduser

4.1.3.15.3 Triggers

Al insertar una nueva fila en la tabla se lanza un evento de creación de una notificación con los datos:

- **emitter**: user1
- **receiver**: user2
- **type**: 'favorite'
- **date**: Establecida automáticamente.

4.1.3.16 tripPoints

Esta tabla contiene los puntos que va a tener cada viaje, su orden, localización y demás datos.

4.1.3.16.1 Columnas

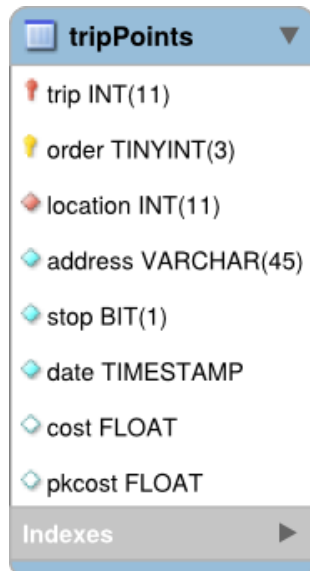


Imagen 50 - Tabla BD - tripPoints

- **trip**: Identificador del viaje.
- **order**: Número de orden del punto.
- **location**: Identificador de la localización del punto.
- **address**: Dirección exacta de donde se realizará la parada.
- **stop**: Si se va a realizar parada o no en este punto.
- **date**: Fecha de salida del punto.
- **cost**: Coste de pasajero desde el punto anterior.
- **pkcost**: Coste de paquete desde el punto anterior.

4.1.3.16.2 Claves extranjeras

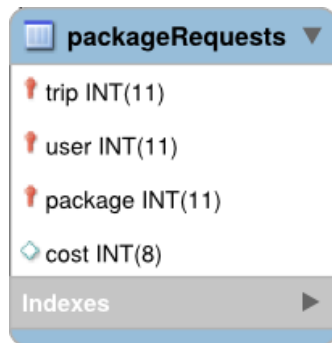
Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_tripPoints_1</i>	trip	trip	idtrip
<i>fk_tripPoints_2</i>	location	location	idlocation

4.1.3.17 packageRequests

Esta tabla contiene las solicitudes de enviar un paquete en un viaje.

4.1.3.17.1 Columna



- **trip**: Identificador del viaje.
- **user**: Usuario que realiza la solicitud.
- **package**: Paquete a transportar.
- **cost**: Coste total de transportar el paquete.

Imagen 51 - Tabla BD - packageRequests

4.1.3.17.2 Claves extranjeras

Esta tabla cuenta con las siguientes claves extranjeras:

NOMBRE DE CLAVE	TABLA REFERENCIADA	COLUMNA	COLUMNA REFERENCIADA
<i>fk_packageRequests_1</i>	trip	trip	idtrip
<i>fk_packageRequests_2</i>	user	user	iduser
<i>fk_packageRequests_3</i>	package	package	idpackage

4.2 Diseño de interfaz

En este apartado se encuentran los bocetos y esquemas de la interfaz de usuario que tendrá la aplicación web.

4.2.1 Área principal

Este área será la estructura básica que tendrá toda la aplicación web, y consta de tres partes diferenciadas, que veremos en mayor detalle a continuación.

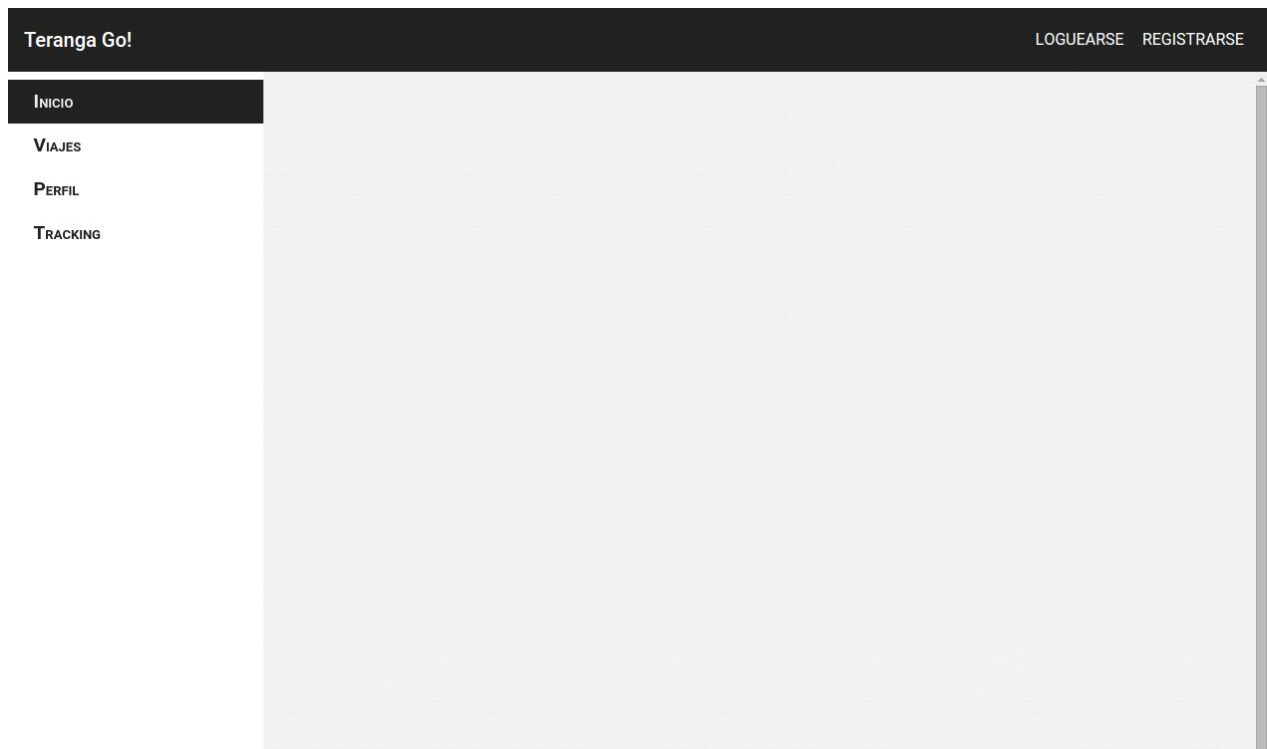


Imagen 52 - Diseño interfaz – Área principal

4.2.1.1 Barra superior

En la zona superior se encuentra una barra con el nombre de la aplicación, así como de botones asociados a la sesión del usuario.



Imagen 53 - Diseño interfaz – Barra superior

En caso de no haber iniciada una sesión, la barra mostrará botones para poder acceder a la página de registro de usuario o abrir un pop-up para introducir los datos para un inicio de sesión.

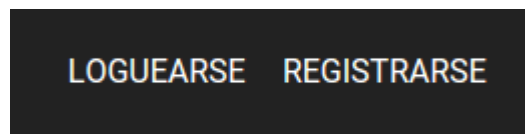


Imagen 54 - Diseño interfaz – Barra superior sin sesión iniciada

Cuando el usuario se encuentra iniciado dentro de la aplicación, los botones anteriores desaparecen, sustituyéndose por dos botones:



Imagen 55 - Diseño interfaz – Barra superior con sesión iniciada

El botón de la derecha permite al usuario acceder a su perfil o cerrar la sesión:

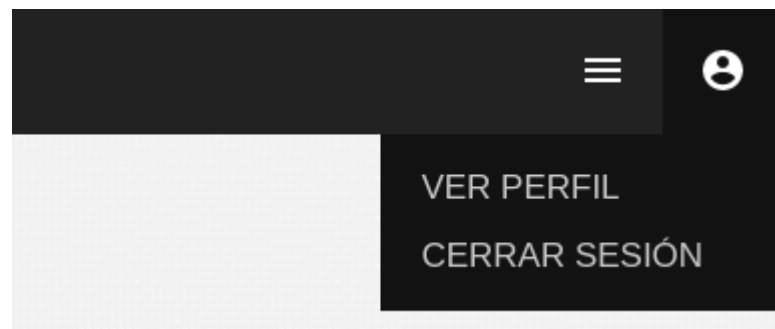


Imagen 56 - Diseño interfaz – Botón de perfil

El botón de la izquierda muestra al usuario sus notificaciones:

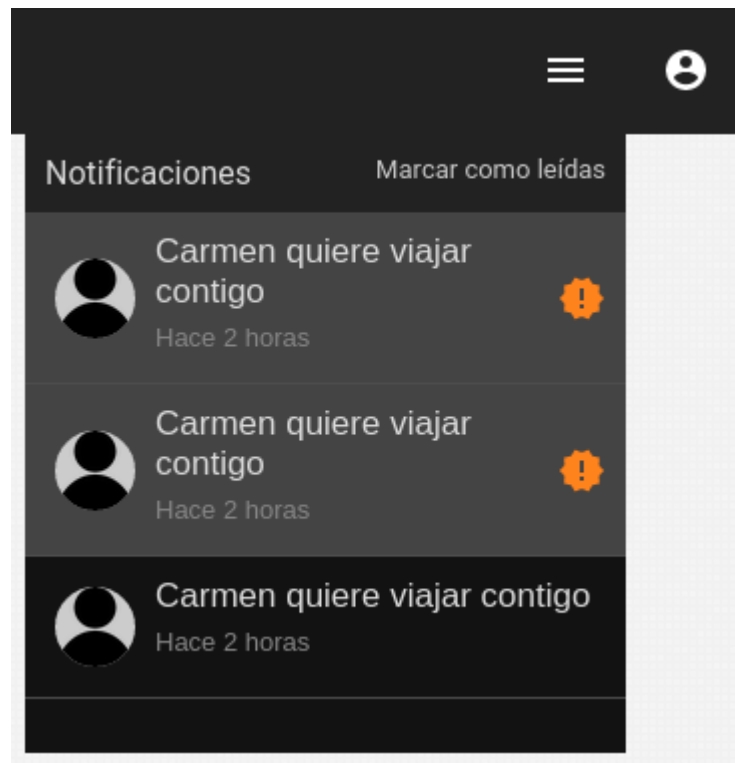


Imagen 57 - Diseño interfaz – Botón de notificaciones

En caso de tener notificaciones no leídas, se muestran sobre el botón de forma resaltada:

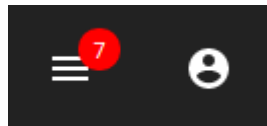


Imagen 58 - Diseño interfaz –Notificaciones no leídas

Al abrir el menú desplegable de las notificaciones se mostrarán de forma resaltada aquellas notificaciones no leídas frente a las leídas.



Imagen 59 - Diseño interfaz – Notificaciones leídas vs. no leídas

Existen notificaciones de diferentes temas, que informarán al usuario de las diferentes interacciones que tiene con otros usuarios, como es el cambio de estado de sus solicitudes de viaje, que haya sido marcado como favorito, etc.

4.2.1.2 Menú lateral

En el lateral izquierdo se localiza el menú de la aplicación, con un enlace a las páginas más usadas del sistema:



Imagen 60 - Diseño interfaz- Menú lateral

- **Inicio:** Te redirige a la página principal de la aplicación.
- **Perfil:** Te enlaza a la página de tu perfil en caso de tener una sesión iniciada o a la página de registro en caso contrario.
- **Viajes:** Te permite buscar entre los viajes existentes.
- **Tracking:** Te lleva a una página para controlar la localización y estado de tus paquetes.

4.2.1.3 Área central

En éste área se visualizarán las distintas páginas de la aplicación.

4.2.2 Zona de perfil / registro


Se puede acceder a estas páginas tanto desde el menú lateral (“*Perfil*”) o desde los botones de la barra superior (“*registrarse*” si no hay sesión iniciada y “*Ver perfil*” en caso contrario)

4.2.2.1 Registro

Cuando un usuario no ha iniciado sesión, al entrar en la página de perfil o al hacer click en el botón de registrarse, es conducido a la página de registro, en donde podrá introducir sus datos para convertirse en usuario registrado de la página.

¿Todavía no te has registrado? ¡Hazlo gratis!

¿Ya tienes una cuenta? [LOGUEATE](#)



Conductor ☐

Nombre

Apellidos

email

telefono

Fecha de nacimiento

dd/mm/aaaa

Contraseña

Repetir contraseña

[GUARDAR](#)

Imagen 61 - Diseño interfaz – Página de registro

Para poder registrarse, es obligatorio rellenar los campos de nombre, apellidos, correo electrónico, teléfono, fecha de nacimiento y contraseña.

Como opcionales se encuentran los campos de introducir una foto de perfil o indicar si es conductor.

4.2.2.2 Inicio de sesión

Para poder entrar como usuario registrado es necesario rellenar el siguiente formulario de logueo, que aparece como un pop-up sobre la página.

Un formulario de inicio de sesión con un campo de texto para 'Email', un campo de texto para 'Contraseña' y un botón naranja con el texto 'INICIAR'.

Imagen 62 - Diseño interfaz – Formulario de logueo

4.2.2.3 Perfil propio

Una vez registrados podemos consultar nuestro perfil:

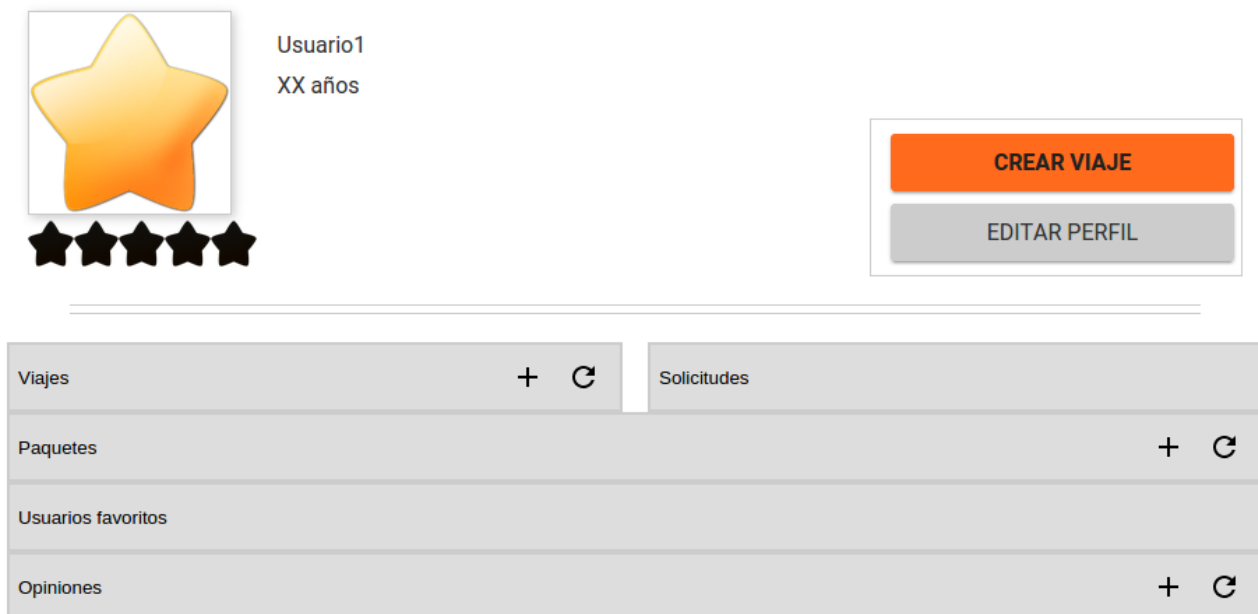
Una interfaz de perfil de usuario. A la izquierda hay un avatar con una estrella amarilla y cinco estrellas negras debajo. Al lado del avatar se indica 'Usuario1' y 'XX años'. A la derecha hay dos botones: uno naranja 'CREAR VIAJE' y uno gris 'EDITAR PERFIL'. Debajo de esto hay una barra de navegación con pestañas: 'Viajes' (con un icono de '+' y un icono de 'recargar'), 'Solicitudes', 'Paquetes' (con un icono de '+' y un icono de 'recargar'), 'Usuarios favoritos' y 'Opiniones' (con un icono de '+' y un icono de 'recargar').

Imagen 63 - Diseño interfaz – Perfil propio

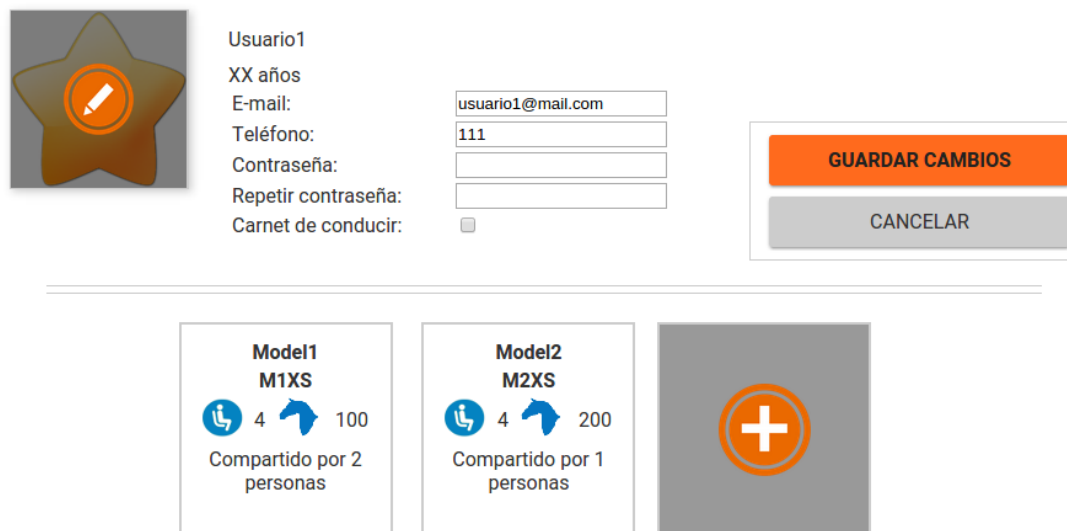
En la zona superior se puede encontrar información sobre el nombre, apellidos y edad del usuario, así como su foto de perfil en caso de tener y la puntuación de 1 a 10 representada gráficamente en caso de ser conductor.

Además, el perfil cuenta con una serie de desplegables que le permiten acceder a información de sus viajes, solicitudes, paquetes y usuarios favoritos. Cada uno de estos desplegables contiene módulos de su tipo correspondiente con la información básica (ver información **de módulos 4.2.5**)

4.2.2.4 Editar perfil

Desde la zona de perfil se puede seleccionar la opción “Editar perfil” que permite editar los campos de si es conductor, email. Teléfono, contraseña e imagen.

Además, permite añadir o modificar los coches creados. (Ver **módulo 4.2.5.3**)



El formulario de edición de perfil está dividido en tres secciones principales. A la izquierda, hay un icono de una estrella con un lápiz. Al lado, se muestran los campos de texto para: 'Usuario1', 'XX años', 'E-mail:' (con el valor 'usuario1@mail.com'), 'Teléfono:' (con el valor '111'), 'Contraseña:', 'Repetir contraseña:' y 'Carnet de conducir:' (con un checkbox). A la derecha, hay dos botones: 'GUARDAR CAMBIOS' (naranja) y 'CANCELAR' (gris). Debajo de esta sección, hay una barra horizontal que separa el formulario de una lista de vehículos. La lista muestra dos vehículos: 'Model1 M1XS' (con un icono de coche y un icono de persona, y los valores '4' y '100') y 'Model2 M2XS' (con un icono de coche y un icono de persona, y los valores '4' y '200'). Ambos vehículos tienen el texto 'Compartido por 2 personas' y 'Compartido por 1 personas' respectivamente. A la derecha de la lista, hay un icono de una cruz blanca dentro de un círculo naranja.

Imagen 64 - Diseño interfaz – Editar perfil

4.2.2.5 Perfil de otro usuario

Al visitar el perfil de otro usuario pueden verse información básica, así como sus viajes pendientes y opiniones recibidas en caso de tenerlas.

Diseño previo no disponible

En caso de tratarse de un conductor el usuario en cuestión, el perfil mostrará la valoración que ha obtenido de otros usuarios tras sus viajes.



Imagen 65 - Diseño interfaz - Estrellas

También se mostrará en el lateral un desplegable mostrando aquellos viajes que va a realizar de los que el usuario es conductor.

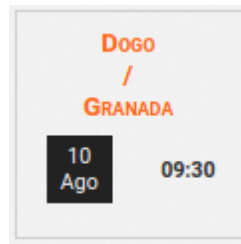


Imagen 66 - Diseño interfaz – Viajes de otro usuario

En caso de no ser conductor, estos campos permanecerán ocultos.

4.2.3 Viajes

4.2.3.1 Creación de viajes

Ruta
Ruta1

De: Granada Dirección

A: Portos Dirección

Fecha dd/mm/aaaa Hora --:--

Coche

Asientos libres 1

Precio

Asiento x 13€
Paquete x 10€

TRAYECTO COMPLETO

Granada			Salida	10 Jun	10:30
Motril			Retardo	--:--	Salida 10 Jun 10:30
Gibraltar			Retardo	--:--	Salida 10 Jun 10:30
Portos					

GUARDAR **CANCELAR**

Imagen 67 - Diseño interfaz – Creación de viajes

Esta página permite crear un nuevo viaje, permitiéndote elegir entre las posibles rutas que ofrece la aplicación, localizaciones de salida y destino, si se va a permitir llevar animales o paquetes, que coche usar, etc.

La aplicación mostrará una lista con las paradas intermedias de la ruta para que el usuario pueda indicar si va a realizar parada en cada una de ellas y, en caso afirmativo, lugar y tiempo de estancia en ella.

Motril			Retardo	--:--	Salida	10 Jun	10:30
--------	--	--	---------	-------	--------	--------	-------

Imagen 68 - Diseño interfaz – Localización sin parada

Gibraltar			Retardo	--:--	Salida	10 Jun	10:30
-----------	--	--	---------	-------	--------	--------	-------

Imagen 69 - Diseño interfaz – Localización con parada

4.2.3.2 Búsqueda de viajes

Encuentra el viaje que necesitas

De:	A:
Origen	Destino
Selecione una opción de la lista	Selecione una opción de la lista
Cuando lo necesitas	
Desde: 09/08/2015	Hasta: 09/08/2015
  	  
BUSCAR	

Este módulo permite seleccionar un origen y destino y un rango de fechas, así como las preferencias de viaje para mostrar al usuario aquellos viajes que encajen en su solicitud. Estos viajes, en caso de existir, se mostrarán en una lista a la cual podrán aplicarse filtros de ordenación diversos.

Viajes de a Ordenar por: Precio ascendente ▼		
 ★★★★★ Usuario1	Granada - Dogo 13 Jul 12:38 1 plazas libres	273.47€ por persona VER
 ★★★★★ Usuario1	Granada - Dogo 13 Jul 12:40 3 plazas libres	127€ por persona VER

Imagen 70 - Diseño interfaz – Lista de viajes

En cada viaje se mostrará información básica sobre el conductor, el precio por persona y datos de salida del viaje.

4.2.3.3 Información de viaje

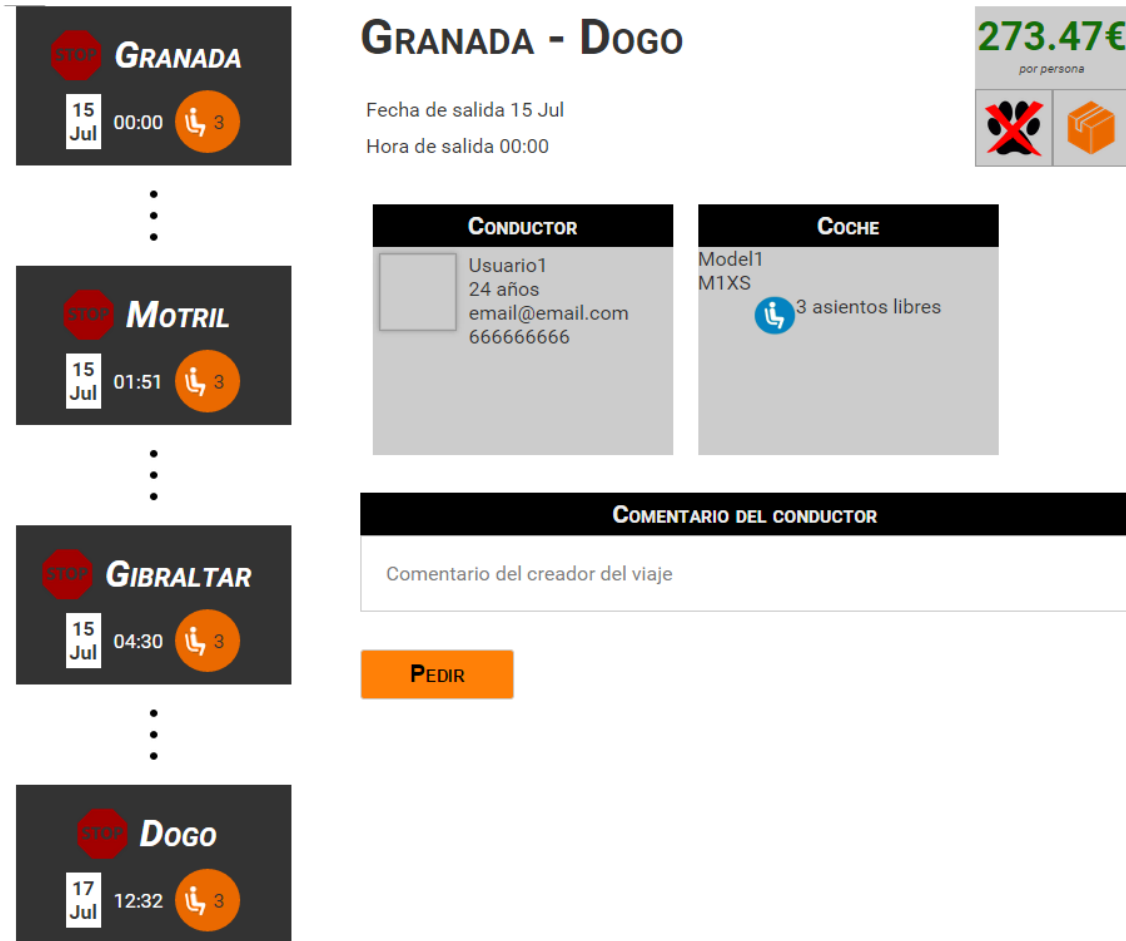


Imagen 71 - Diseño interfaz – Información de viaje

Esta página muestra toda la información detallada de un viaje, dividido en varias zonas:

- El lateral izquierdo muestra todas las localizaciones intermedias por las que se va a pasar en el viaje, así como si se va a realizar parada en ellas y cuando es la fecha de salida y asientos libres.
- En la esquina superior derecha se muestra información del precio y preferencias.
- En la parte central se muestra la información general del viaje: Información del conductor y coche en el que se realizará el viaje y comentario del conductor sobre ese viaje en concreto.

4.2.3.4 Solicitud de viaje

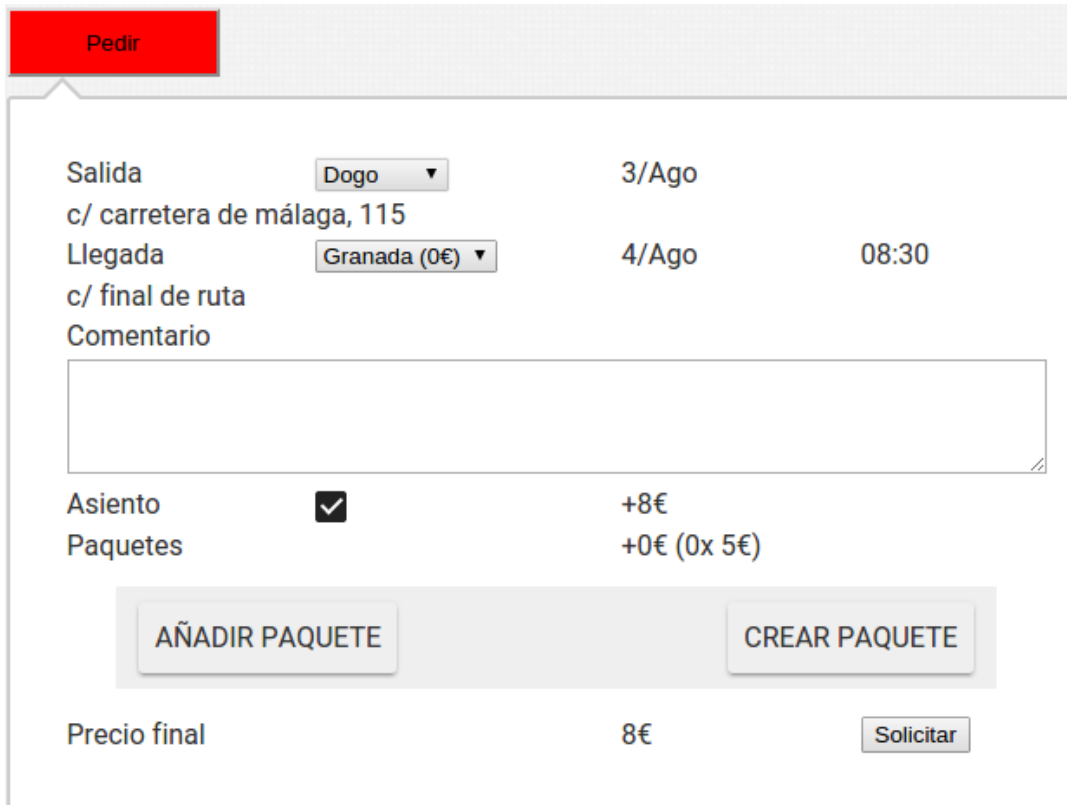
Al mirar la información de un viaje puedes solicitar, si es posible, unirse como pasajero a él. Para ello se debe rellenar el siguiente formulario:



Formulario de solicitud de viaje sin paquete. El formulario está dividido en secciones. En la parte superior, hay un botón rojo que dice "Pedir". Debajo, hay un campo "Salida" con un menú desplegable que muestra "Granada" y un campo de fecha que muestra "3/Ago". A continuación, hay un campo de texto que dice "c/ carretera de Málaga, 115". Luego, hay un campo "Llegada" con un menú desplegable que muestra "Dogo (4€)" y un campo de fecha que muestra "4/Ago" y un campo de hora que muestra "08:30". A continuación, hay un campo de texto que dice "c/ final de ruta". Luego, hay un campo "Comentario" con un área de texto grande. En la parte inferior, hay un campo "Asiento" con un menú desplegable que muestra "✓" y un campo de precio que muestra "+8€". A continuación, hay un campo "Precio final" que muestra "8€" y un botón que dice "Solicitar".

Imagen 72 - Diseño interfaz – Solicitud sin paquete

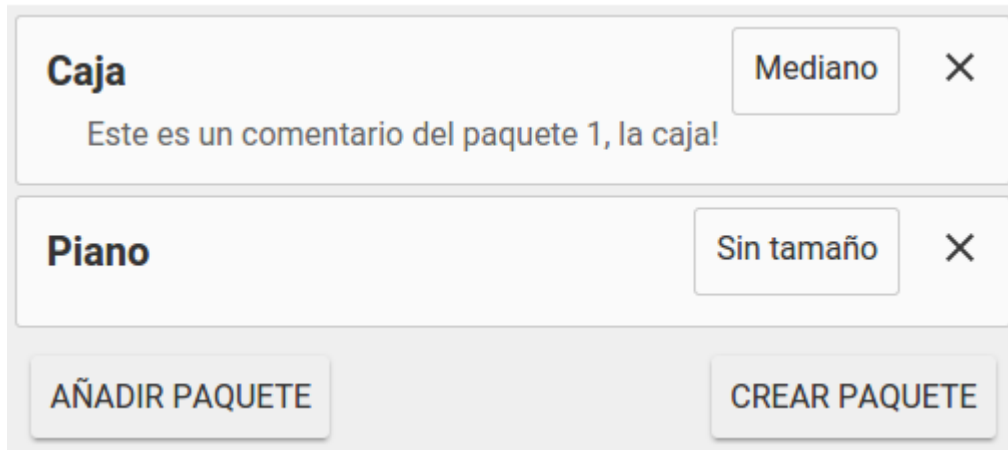
En caso de que el viaje acepte llevar paquetes, la solicitud varía, mostrándose lo siguiente:



Formulario de solicitud de viaje con paquete. El formulario está dividido en secciones. En la parte superior, hay un botón rojo que dice "Pedir". Debajo, hay un campo "Salida" con un menú desplegable que muestra "Dogo" y un campo de fecha que muestra "3/Ago". A continuación, hay un campo de texto que dice "c/ carretera de Málaga, 115". Luego, hay un campo "Llegada" con un menú desplegable que muestra "Granada (0€)" y un campo de fecha que muestra "4/Ago" y un campo de hora que muestra "08:30". A continuación, hay un campo de texto que dice "c/ final de ruta". Luego, hay un campo "Comentario" con un área de texto grande. En la parte inferior, hay un campo "Asiento" con un menú desplegable que muestra "✓" y un campo de precio que muestra "+8€". A continuación, hay un campo "Paquetes" con un menú desplegable que muestra "+0€ (0x 5€)". Debajo de estos campos, hay dos botones: "AÑADIR PAQUETE" y "CREAR PAQUETE". En la parte inferior, hay un campo "Precio final" que muestra "8€" y un botón que dice "Solicitar".

Imagen 73 - Diseño interfaz – Solicitud con paquete

En este último caso, es posible solicitar llevar un paquete, sin que tenga que ir el usuario como pasajero. Para ello se puede crear un nuevo paquete o seleccionar de una lista de los existentes. Tras crear o seleccionar el paquete, este se añadirá a una lista de la solicitud y se actualizará el precio total a pagar.



Caja	Mediano	×
Este es un comentario del paquete 1, la caja!		
Piano	Sin tamaño	×
AÑADIR PAQUETE		
CREAR PAQUETE		


Imagen 74 - Diseño interfaz – Lista de paquetes en solicitud

4.2.4 Paquetes

La aplicación permite crear paquetes para mandarlos en un viaje y llevar un control de ellos mientras están de camino.

4.2.4.1 Información de paquete

VOLVER



Caja
ID: 1

Mediano

Este es un comentario del paquete 1, la caja!

ELIMINAR PAQUETE

Granada

Gibraltar

Motril

Portos

Dogo

Estado: Inactivo

Fecha estimada de salida: 10 Junio a las 10:30

Fecha estimada de llegada: 10 Junio a las 10:30

Última localización: Motril

Imagen 75 - Diseño interfaz – Información paquete

En esta página se podrá ver toda la información de un paquete, así como su localización aproximada en caso de encontrarse en un viaje.

4.2.4.2 Creación de paquete

Desde el perfil o el formulario de solicitud de viaje se puede crear un nuevo paquete en el que se indica datos generales del paquete.



Formulario de creación de nuevo paquete:

- Título:** NUEVO PAQUETE
- Nombre:** nombre de paquete
- Tamaño:** 4 (seleccionado en un slider)
- Descripción:** 0/200
- Usuario receptor y/o DNI, NIF o NIE del receptor:**
- Botones:** CREAR (naranja), CANCELAR (gris)

Imagen 76 - Diseño interfaz – Creación de paquete

4.2.5 Módulos

Para mostrar información básica de diversos elementos de la aplicación se han creado módulos para mostrar esta información de forma clara y entendible.

4.2.5.1 Viaje

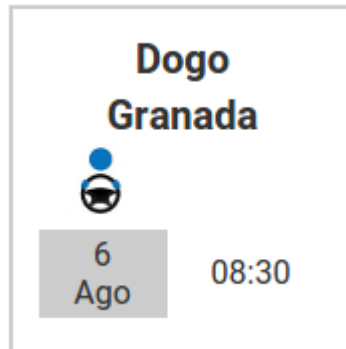


Imagen 77 - Diseño interfaz – Módulo viaje

Muestra información de conductor, salida, destino y fecha de un viaje concreto.

4.2.5.2 Solicitud



Imagen 78 - Diseño interfaz – Módulo solicitud

Muestra información de usuario que hace la solicitud e información de esta.

4.2.5.3 Coche



Imagen 79 - Diseño interfaz – Módulo coche

Muestra información de matrícula, modelo y conductores de un coche.

4.2.5.4 Paquete

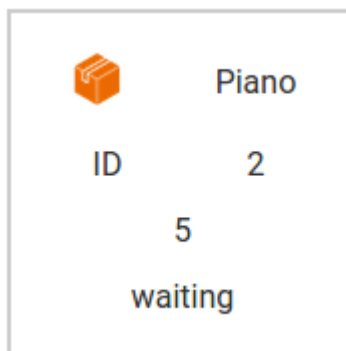


Imagen 80 - Diseño interfaz – Módulo paquete

Muestra información de nombre, ID^[A7], tamaño y estado de un paquete.

4.2.5.5 Usuario favorito



Imagen 81 - Diseño interfaz – Módulo usuario favorito

Muestra información de nombre, puntuación y viajes activos (en caso de tenerlos) de un usuario.

4.2.5.6 Opinión / Comentario

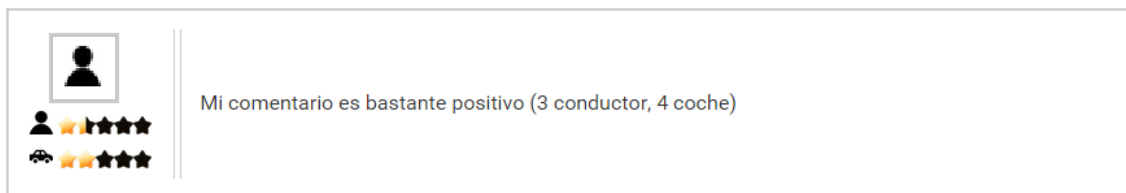


Imagen 82 - Diseño interfaz – Módulo comentario

Muestra información del usuario que realiza el comentario y el comentario en sí.

4.2.6 Responsividad^[B12]

Dado que la página web puede ser visualizada en diversos dispositivos, desde ordenadores con pantallas de gran resolución hasta tablets y smartphones, la *responsividad*^[17] se convierte en una necesidad a la hora de crear la página web.

En nuestro caso, *Polymer*^[5.1.2.2] nos aporta herramientas y módulos para facilitar la creación de una web dinámica y adaptable a las diferentes resoluciones de pantalla.

En el caso del menú lateral, al cargarse la página web en una pantalla de poca resolución este desaparece, pudiendo accederse a él deslizando el dedo o ratón desde el lateral izquierdo hacia la derecha, al igual que en múltiples aplicaciones de móvil. Esto permite dedicar mayor espacio a la parte central de la aplicación.

De manera similar, la barra superior que se muestra en la página web disminuye su tamaño, para ocupar el espacio justo. No obstante, aunque algunas barras superiores que aporta *Polymer*^[5.1.2.2] desaparecen por completo, en nuestra aplicación permanece siempre visible, para poder, o bien iniciar sesión en cualquier momento o bien tener constancia de las notificaciones de la aplicación.

Con respecto al área principal de la aplicación, se ha establecido como máximo un tamaño de cuerpo de *900px*, de modo que no quede desproporcionada en pantallas demasiado grandes. Para pantallas menores, los elementos internos se adaptan al tamaño de esta, siempre con un mínimo por elemento que permita una fácil legibilidad de este.

5. Implementación

5.1 Herramientas utilizadas

A lo largo del desarrollo del proyecto se ha hecho uso de diversas tecnologías, lenguajes de programación, bibliotecas, etc. En este capítulo se habla con detalle acerca de cada una de estas herramientas y de su aplicación que han tenido dentro del proyecto.

5.1.1 Lenguajes utilizados

En este apartado se hablará acerca de los lenguajes usados para la creación del programa en el que se basa el proyecto, tanto aquellos lenguajes de programación, como de etiquetado y estilo.

5.1.1.1 JavaScript ^[B17]

Es un lenguaje de programación interpretado que apareció en el año 1995, desarrollado por *Netscape* con el nombre de *Mocha*, que cambió su nombre a *JavaScript* coincidiendo con el comienzo de la compatibilidad de Java en su navegador.

En 1997 se propuso el lenguaje para que fuera adoptado como estándar de la *European Computer Manufacturers Association*. Poco después se convirtió también en un estándar *ISO* ^[A10].

Aunque tiene diversas aplicaciones, en este proyecto se usará como parte de la página web, utilizándose en el lado del cliente para mejorar la interfaz de usuario y realizar cálculos en el ordenador del usuario, de modo que se aligera la carga de procesamiento que tiene que soportar el servidor web.

Dado que el código *JavaScript* puede ejecutarse localmente en el navegador del usuario sin necesidad de volver a realizar una petición al servidor, el navegador puede responder a las acciones del usuario rápidamente, haciendo una aplicación más eficiente y fluida. Por otra parte, el código *JavaScript* puede detectar acciones de los usuarios que en caso de no usarse serían imposibles de detectar, como pulsaciones de teclado.

Todos los navegadores modernos interpretan el código *JavaScript* integrado en las páginas web. Para interactuar con una página web se provee al lenguaje *JavaScript* de una implementación del *Document Object Model* (*DOM* ^[A12]), descargándose junto con el código *HTML* ^[A11] para ser usado en el navegador.

JavaScript se diseñó con una sintaxis similar al *C*, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo *Java* y *JavaScript* no están relacionados y tienen semánticas y propósitos diferentes.

5.1.1.2 SQL (MySQL) ^[B17]

SQL es un lenguaje de consulta estructurado (*Structured Query Language*) usado para el acceso y manejo de bases de datos relacionales.

En 1970, *IBM* definió el lenguaje *SEQUEL*, que más tarde fue implementado como sistema gestor de bases de datos. Este lenguaje sería el predecesor de *SQL*, pasando a ser el lenguaje por excelencia de los diversos sistemas de gestión de bases de datos a lo largo de los siguientes años, estableciéndose como estándar en el año 1986.

SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones, consiguiéndose las siguientes características entre otras:

- Definir esquemas de relación entre los datos almacenados.
- Hacer consultas de los datos basados en restricciones.
- Especificar restricciones de integridad que deben cumplir los datos.
- Sistemas de permisos.

En este proyecto, este lenguaje se usará para almacenar y tratar los datos referentes a los usuarios, así como para mostrar por pantalla aquellos datos necesarios.

5.1.1.3 HTML5 ^[B11]

HTML comenzó su desarrollo en 1991, donde *Tim Berners-Lee* describió los primeros elementos que formarían parte de este lenguaje y en 1993 fue reconocido como una ampliación del lenguaje *SGML*.

Es un lenguaje de marcado para la creación de páginas web, que hace uso de etiquetas para la distribución de elementos dentro de la página y de referencias para la introducción de elementos como imágenes, vídeos, etc.

Para el formato de la página creada, *HTML* hace una referencia a un archivo *CSS* ^[A13] el cual contiene toda la información de estilo.

En nuestro proyecto, este lenguaje será utilizado como base para la creación de la página web.

5.1.1.4 CSS3^[B10]

CSS es un lenguaje de hojas de estilo propuesto por *Håkon Wium Lie* en 1994 y fue incluido en *World Wide Web Consortium (W3C)* en 1996.

Es usado para dar formato y estilo a estructuras *HTML*, *XHTML* o *XML*, permitiendo variar cada elemento por separado con su *ID*^[A7] único o agrupándolos en clases.

En nuestro proyecto, haremos uso de este lenguaje para dar formato a toda la página web.

5.1.2 Tecnologías usadas

Aquí se detalla la tecnología de desarrollo utilizada en este proyecto.

5.1.2.1 AJAX^[B17]

AJAX (Asynchronous JavaScript And XML)^[A14] es una tecnología de desarrollo web que se ejecuta en navegador del cliente mientras mantiene una comunicación asíncrona con el servidor en segundo plano. Esto permite realizar cambios en la página sin tener que recargarla o solicitar al servidor tan solo un módulo de datos, sin tener que volver a enviar la página completa.

Esta técnica muestra grandes ventajas en su utilización, tales como el ahorro de transmisión de datos, tanto por parte del usuario como del servidor, amenizar el viaje entre páginas del usuario, aportando fluidez y acelerando todo el proceso de mostrado de las páginas.

AJAX es una combinación de otras tecnologías ya existentes:

- *HTML*^[A11] y *CSS*^[A13] para la estructura y diseño de los datos que mostrará.
- *DOM*^[A12] para mostrar e interactuar dinámicamente con la información mostrada.
- *XML*^[A14] para dar formato a los datos transferidos.

En este proyecto se usará una versión modificada de *AJAX*: *AJAJ*, el cual usa *JSON*^[A15] como formato ligero para el intercambio de datos, sustituyendo a *XML*^[A14]. Esto se debe a las ventajas que presenta ante *AJAX*, ya que resulta más ligero, acelerando el proceso de transferencia de datos y reduciendo el tráfico de red usado. Además presenta un formato mucho más simplificado y legible, por lo que facilita el trabajo con estos documentos.

5.1.2.2 Polymer^[B9]

Polymer es una biblioteca de componentes web que permite encapsular elementos personalizados que extienden de *HTML*^[A11] de forma sencilla. Actualmente se encuentra en su versión 1.0.

Creada por Google recientemente, para la unificación de su diseño, cuenta con una *API*^[A5] que permite crear páginas web de forma rápida a base de módulos.

Además de permitir crear tus propios módulos, *Polymer* cuenta actualmente con una gran variedad de módulos que aportan la funcionalidad básica y conexión con la *API*^[A5] de Google y con un diseño claro y limpio, que se pueden añadir a tu web con apenas una línea de código. Estos módulos no serán descargados por el usuario a menos que vayan a ser utilizados, por lo que el peso de la web depende de los módulos usados, sin sobrecargar innecesariamente esta.

Polymer se basa en el comportamiento de encapsulamiento, expuesto teóricamente por los *Web Components*^[T2]. Esta se basa en el desarrollo de una web a base de componentes separados y exportables, que permiten la modularización y limpieza de código.

En este proyecto se utilizará *Polymer* como base junto con *HTML*^[A11] para la creación de la página, usando módulos ya creados y creando módulos personalizados para aquellas tareas que no estén cubiertas por los módulos anteriores.

5.1.3 Servidor

Aquí se detalla el entorno utilizado para la capa de servidor del proyecto.

5.1.3.1 NodeJS con express^{[B7] [B8]}

NodeJS es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor que utiliza el lenguaje *JavaScript*^[5.1.1.1]. Desarrollado en 2009 por *Ryan Dahl*.

Express es un framework web creado especialmente para *NodeJS*, que permite crear de forma fácil una *API*^[A5] para trabajar con los datos de una base de datos.

Juntando estas dos tecnologías se puede crear una *API REST*^[A3]. Esta *API*^[A5] es un conjunto de funciones a las que se accede a través de una dirección web o *URL* en las que incluimos los datos de la consulta a realizar.

En nuestro proyecto usaremos la *API REST*^[A3] para realizar consultas a la base de datos por medio de una estructura definida de *URL*, por lo que podrá usarse tanto en la aplicación web como en una futura aplicación móvil sin necesidad de cambiar nada.

5.1.4 Programas externos

En este apartado se explican aquellos programas cuyo uso ha sido necesario o ha facilitado la tarea de creación del proyecto.

5.1.4.1 MySQL Workbench^[B13]

MySQL Workbench es una aplicación para el diseño de bases de datos que integra desarrollo *SQL*, administración, diseño creación y mantenimiento de una base de datos en un solo entorno.

Sucesora de la aplicación *DBDesigner4*, fue publicada en 2005 y actualmente pertenece a *Oracle*. Cuenta con una versión de código abierto y otra comercial con funcionalidades exclusivas.

En este proyecto se usará la versión de código libre para la administración de la base de datos.

5.1.4.2 Postman^[B12]

Postman es una aplicación que permite testear una *API REST*^[A3], permitiendo crear y mandar cualquier solicitud de *HTTP* de forma fácil y sencilla.

Estas solicitudes son almacenadas por la aplicación permitiendo recuperarlas y modificarlas de forma sencilla. Además permite crear scripts para la generación de peticiones, lo que permite crear una batería de pruebas para comprobar el correcto funcionamiento de nuestra *API REST*^[A3].

En este proyecto se usa *Postman* como herramienta de testeo de la *API REST*^[A3] creada con *NodeJS* y *Express*. Esto permite acelerar la tarea de creación de consultas y solicitudes a la base de datos.

5.2 Fases del desarrollo

En este apartado se detallarán cada una de las distintas fases del desarrollo del programa, analizando los problemas hallados en cada una de ellas y la solución propuesta.

5.2.1 Desarrollo de la base de datos^[B17]

En primer lugar se tenía que decidir el método de almacenamiento de los datos. Para ello se barajaron dos opciones principalmente: *MongoDB* y *MySQL*. Ambas presentaban ventajas e inconvenientes a la hora de su uso en la aplicación, por lo que se realizó una tabla como la que se muestra a continuación para determinar cual opción usar:

	MONGODB	MySQL
<i>TIPO</i>	BigData (no relacional)	Relacional
<i>ESCALABILIDAD</i>	Buena	Mala
<i>FACILIDAD DE CONSULTAS</i>	Bajo	Medio/Alto
<i>VELOCIDAD DE ESCRITURA</i>	Mayor	Menor
<i>VELOCIDAD DE LECTURA</i>	Mayor	Menor
<i>VELOCIDAD DE AGREGACIÓN</i>	Menor	Mayor
<i>INTEGRIDAD RELACIONAL</i>	Menor	Mayor
<i>PREFERENCIAS</i>	Preferido por el programador	Preferido por la empresa

Tabla 2 - Comparativa MongoDB vs MySQL

Tras ver la tabla anterior, aunque *MongoDB* tenga mejores resultados, se ha optado por el uso de *MySQL* por petición de la empresa, que ya cuenta con una base de datos de este tipo.

Una vez decidido el tipo de base de datos se procedió a crear la estructura necesaria para almacenar los datos de forma correcta. Se barajaron diversas opciones intentando reducir lo máximo posible el número de tablas y relaciones, pero sin perder de vista la optimización.

Un ejemplo de esto es el siguiente: Se tiene un usuario con diversos datos, entre los que se encuentra el código de confirmación de e-mail para su registro. Existen dos formas de abarcar esto:

- Creación de una única tabla usuario con todos los datos, incluido el código de confirmación.
- Creación de una tabla usuario y otra que relacione usuario con código de confirmación.

La segunda opción permite el borrado de estos datos cuando el usuario se ha registrado, liberando memoria en vez de tener que mantener creado el campo de código en la tabla de la primera opción.

Considerando estas mejoras, se creó el diseño final de la base de datos que se muestra en el **apartado 4.1.2** de esta documentación.

5.2.2 Creación de la API REST^[B8]

Para la solicitud de datos al servidor se escogió hacer uso de una *API REST*, permitiendo su uso sin cambios extendido a otras aplicaciones, como puede ser una aplicación móvil o un programa de escritorio, aparte de la aplicación web a crear.

Haciendo uso de Express se creó la *API REST* necesaria. Para ello se siguieron varios pasos:

- Creación de la sentencia *SQL* necesaria con ayuda del programa *MySQL Workbench*. Este programa permite crear y comprobar las sentencias *SQL* de una base de datos, facilitando la tarea de creación de la API^[A5].
- Creación de la *URL* necesaria. En este paso se usará el programa *Postman* con el cual se podrán realizar de forma fácil peticiones de *URL*, por lo que se podrá comprobar si la consulta es correcta.

Este proceso se repite para cada una de las distintas consultas, creaciones y eliminaciones de datos de la aplicación.

5.2.2.1 Uso de la API

Para hacer uso de la *API REST*^[A3] se realizan peticiones en forma de *URL* al servidor. La estructura general de una solicitud será la siguiente:

[URLRoot]/api/v1/[endPoint]

Donde:

- **[URLRoot]**: Dirección donde está alojada la página.
- **[endPoint]**: Petición a la *API REST*^[A3].

5.2.2.1.1 Objetos de una petición

Al realizar una petición, esta toma datos de diferentes lugares, pudiendo ser:

- **Body**: Conocido también como método *POST*. Toma los valores de un formulario existente en la página en el momento de realizar la petición. Esto se utiliza para peticiones que van a modificar la *BD*^[A6] o datos que no quieren mostrarse públicamente, como puede ser la contraseña.
- **Query**: Conocida también como método *GET*. Atributos existentes dentro de la *URL*. Se utilizan para realizar consultas a la *BD*^[A6] con datos que pueden ser vistos públicamente.
 - Para añadir estos datos debe introducirse primero el carácter '?' y a continuación todos los datos y cada atributo va separado por el carácter '&'.
 - Cada parámetro está escrito de la forma '*a=b*' donde '*a*' es el nombre del atributo y '*b*' su valor.
- **Parámetros**: Introducidos también en la *URL* son cadenas alfanuméricas precedidas por el carácter ':'

5.2.2.1.2 Tipos de peticiones

Dependiendo del método usado, las llamadas de *API REST*^[A3], estas se dividen en:

- **GET**: Solicita datos de la *BD*^[A6] al servidor.
- **POST**: Introduce nuevos datos a la *BD*^[A6] del servidor.
- **DELETE**: Elimina datos de la *BD*^[A6] del servidor.
- **PUT**: Modifica datos de la *BD*^[A6] del servidor.

5.2.2.2 Peticiones de la aplicación

A continuación se muestra una tabla con un listado de las peticiones que realiza la aplicación actualmente, indicando el método que utiliza, la ruta, parámetros y una breve descripción de esta.

RUTA	MÉTODO	PARÁMETROS	DESCRIPCIÓN
/auth	POST	<ul style="list-style-type: none"> • Body <ul style="list-style-type: none"> ○ email: <i>Email</i> ○ contraseña: <i>String</i> • Query <ul style="list-style-type: none"> ○ logout: <i>Boolean</i> • Parámetros 	<p>Autentica a un usuario en la aplicación, permitiéndole iniciar sesión.</p> <p>En caso de que el atributo “logout” se encuentre presente se cerrará la sesión.</p>
/users	POST	<ul style="list-style-type: none"> • Body <ul style="list-style-type: none"> ○ email: <i>Email</i> ○ contraseña: <i>String</i> ○ * • Query • Parámetros 	Crea un nuevo usuario con los parámetros introducidos
	GET	<ul style="list-style-type: none"> • Body • Query <ul style="list-style-type: none"> ○ * • Parámetros 	Devuelve una lista de todos los usuarios existentes en la BD
/users/:iduser	GET	<ul style="list-style-type: none"> • Body • Query <ul style="list-style-type: none"> ○ * • Parámetros <ul style="list-style-type: none"> ○ iduser: <i>ID</i> 	Devuelve la información acerca de un usuario.
	PUT	<ul style="list-style-type: none"> • Body <ul style="list-style-type: none"> ○ * • Query • Parámetros <ul style="list-style-type: none"> ○ iduser: <i>ID</i> 	Modifica los datos de un usuario.
/users/:iduser/cars	GET	<ul style="list-style-type: none"> • Body • Query • Parámetros <ul style="list-style-type: none"> ○ iduser: <i>ID</i> 	Devuelve los coches de los cuales el usuario es conductor.

/users/:iduser/assessments	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> iduser: <i>ID</i> 	Devuelve todas las opiniones realizadas a un usuario.
/users/:iduser/packages	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> iduser: <i>ID</i> 	Devuelve todos los paquetes creados por un usuario.
/users/:iduser/trips	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> iduser: <i>ID</i> 	Devuelve todos los viajes creados por el usuario.
/users/:iduser/favorites	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> iduser: <i>ID</i> 	Devuelve todos los usuarios que el usuario indicado ha marcado como favoritos.
/users/:iduser/requests	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> iduser: <i>ID</i> 	Devuelve todas las solicitudes del usuario (tanto las que ha creado él como las que le han enviado).
/users/:iduser/notifications	GET	<ul style="list-style-type: none"> Body Query <ul style="list-style-type: none"> since: <i>UTC DateISOString</i> Parámetros <ul style="list-style-type: none"> iduser: <i>ID</i> 	Devuelve las notificaciones que tiene un usuario. Si se especifica el atributo “since” se enviarán solo las notificaciones posteriores a esa fecha.
/favorites	GET	<ul style="list-style-type: none"> Body Query Parámetros 	Obtiene todos los favoritos de todos los usuarios.
	POST	<ul style="list-style-type: none"> Body <ul style="list-style-type: none"> user1: <i>ID</i> user2: <i>ID</i> Query Parámetros 	Indica que el usuario con ID ‘user1’ ha añadido como favorito al usuario con ID ‘user2’.
	DELETE	<ul style="list-style-type: none"> Body <ul style="list-style-type: none"> user1: <i>ID</i> user2: <i>ID</i> Query Parámetros 	Elimina la relación de favorito del usuario con ID ‘user1’ hacia el usuario con ID ‘user2’

/favorites/:iduser	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> iduser: <i>ID</i> 	Devuelve los usuarios que ha agregado como favoritos el usuario con ID ‘ <i>iduser</i> ’
/routes	GET	<ul style="list-style-type: none"> Body Query <ul style="list-style-type: none"> list: <i>Boolean</i> Parámetros 	Devuelve todas las rutas posibles existentes en la aplicación. Si se encuentra el atributo ‘ <i>list</i> ’ devuelve una lista resumida de los datos.
/routes/:idroute	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> idroute: <i>ID</i> 	Devuelve la información acerca de la ruta con ID ‘ <i>idroute</i> ’
	DELETE	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> idroute: <i>ID</i> 	Elimina de la BD la ruta con ID ‘ <i>idroute</i> ’
/trips	POST	<ul style="list-style-type: none"> Body <ul style="list-style-type: none"> wps: <i>Array({location: ID, address: String, stop: Boolean, date: UTC DateISOString, cost: Float, pkcost: Float})</i> animals: <i>Boolean/Null/None</i> packages: <i>Boolean/Null/None</i> Query Parámetros 	Añade un nuevo viaje a la BD con los datos especificados.
	GET	<ul style="list-style-type: none"> Body Query <ul style="list-style-type: none"> origin: <i>ID(location)</i> destination: <i>ID(location)</i> offset: <i>Integer/None</i> limit: <i>Integer/None</i> from: <i>UTC DateISOString/None</i> to: <i>UTC DateISOString/None</i> animals: <i>Boolean/Null/None</i> packages: <i>Boolean/Null/None</i> Parámetros 	Devuelve una lista de viajes que se ajusten a los criterios determinados.
/trips/:idtrip	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> idtrip: <i>ID</i> 	Devuelve la información del viaje con ID ‘ <i>idtrip</i> ’

/cars	POST	<ul style="list-style-type: none"> • Body <ul style="list-style-type: none"> ○ iduser: ID ○ seats: Integer ○ model: String ○ hp: Integer • Query • Parámetros 	Crea un nuevo coche con los datos introducidos.
	GET	<ul style="list-style-type: none"> • Body • Query • Parámetros 	Devuelve una lista de todos los coches existentes en la BD.
/cars/:idcar	GET	<ul style="list-style-type: none"> • Body • Query • Parámetros <ul style="list-style-type: none"> ○ idcar: <i>ID</i> 	Devuelve la información del coche con ID ' <i>idcar</i> '
	PUT	<ul style="list-style-type: none"> • Body <ul style="list-style-type: none"> ○ seats: <i>Integer/None</i> ○ model: <i>String/None</i> ○ hp: <i>Integer/None</i> • Query • Parámetros <ul style="list-style-type: none"> ○ idcar: <i>ID</i> 	Crea un nuevo coche en la BD.
	DELETE	<ul style="list-style-type: none"> • Body • Query • Parámetros <ul style="list-style-type: none"> ○ idcar: <i>ID</i> 	Elimina el coche con ID ' <i>idcar</i> ' de la BD.
/request/travel	POST	<ul style="list-style-type: none"> • Body <ul style="list-style-type: none"> ○ user: <i>ID</i> ○ trip: <i>ID</i> ○ pointA: <i>Integer(Order)</i> ○ pointB: <i>Integer(Order)</i> ○ travels: <i>Boolean</i> ○ cost: <i>Float</i> ○ packages: <i>Array({package: ID, cost: Float})</i> • Query • Parámetros 	Crea una nueva solicitud para que el usuario con ID ' <i>user</i> ' viaje en el viaje con ID ' <i>trip</i> '.
/request/accept	POST	<ul style="list-style-type: none"> • Body • Query <ul style="list-style-type: none"> ○ user: <i>ID</i> ○ trip: <i>ID</i> • Parámetros 	Acepta una solicitud de viaje del usuario con ID ' <i>user</i> ' para el viaje con ID ' <i>trip</i> '

/notifications	GET	<ul style="list-style-type: none"> Body Query <ul style="list-style-type: none"> since: <i>UTC DateISOString</i> Parámetros 	Devuelve todas las notificaciones existentes en la BD.
	PUT	<ul style="list-style-type: none"> Body <ul style="list-style-type: none"> emitter: <i>ID(User)</i> receiver: <i>ID(User)</i> notifications: <i>Array(ID(Notification))</i> Query Parámetros 	Marca las notificaciones indicadas como leídas.
/locations	GET	<ul style="list-style-type: none"> Body Query <ul style="list-style-type: none"> list: <i>Boolean</i> route: <i>ID/None</i> Parámetros 	Devuelve la lista de localizaciones de la ruta de ID ' <i>route</i> '. En caso de especificar el parámetro ' <i>list</i> ' se devolverá una lista de datos reducida.
/locations/:idlocation	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> idlocation: <i>ID/"random"</i> 	Devuelve la información acerca de la localización con ID ' <i>idlocation</i> '.
/packages	GET	<ul style="list-style-type: none"> Body Query Parámetros 	Devuelve todos los paquetes existentes en la BD.
	POST	<ul style="list-style-type: none"> Body <ul style="list-style-type: none"> name: <i>String</i> size: <i>Integer</i> emitter: <i>ID(User)</i> receiver: <i>ID(User)/None</i> Query Parámetros 	Crea un nuevo paquete con los datos introducidos.
/packages/:idpackage	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> idpackage: <i>ID</i> 	Devuelve la información del paquete con ID ' <i>idpackage</i> '
/config	GET	<ul style="list-style-type: none"> Body Query Parámetros 	Devuelve la configuración de la aplicación.

/config/:param	GET	<ul style="list-style-type: none"> Body Query <ul style="list-style-type: none"> KmCost: <i>Boolean</i> KmPkCost: <i>Boolean</i> version: <i>Boolean</i> name: <i>Boolean</i> timezone: <i>Boolean</i> Parámetros <ul style="list-style-type: none"> param: "trips"/"api"/"db" 	Devuelve la configuración establecida para el parámetro indicado.
/actions	GET	<ul style="list-style-type: none"> Body Query Parámetros 	Muestra las opciones de administración disponibles.
/actions/:action	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> action: "generate" 	Muestra la opción de administración indicada.
/actions/:action/:mode	GET	<ul style="list-style-type: none"> Body Query Parámetros <ul style="list-style-type: none"> action: "generate" mode: "car"/"user"/"route"/"trip" 	Ejecuta el modo indicado de la opción de administración indicada.

Tabla 3 - Documentación de API REST

5.2.3 Estructuración de la aplicación web^[B17]^[B9]

Para la creación de la aplicación web se barajaron diversas opciones. Entre ellas *WordPress* y *Polymer*.

WordPress es un *CMS* (Content Management System) o sistema gestor de contenidos, enfocado a la creación de sitios web. Es utilizado especialmente para páginas web con una estructura cronológica que se actualiza regularmente. Se desarrolla en *PHP* y cuenta con una gran comunidad de desarrolladores que pone a disposición de cualquiera complementos y plantillas para realizar determinadas tareas en la web.

WordPress es, en un inicio, un sistema de publicación web basado en entradas ordenadas por fecha, por lo que no encaja con la estructura que se desea en la aplicación de este proyecto.

Por otro lado, se encuentra *Polymer*, la biblioteca utilizada actualmente por Google para unificar y simplificar sus webs. Esta biblioteca permite crear de forma sencilla módulos independientes que poder añadir en cualquier lugar de la aplicación. Esto permite simplificar y modularizar el código.

Además, *Polymer* cuenta con un amplio repertorio de elementos y módulos dinámicos y con un estilo ya definido que se pueden añadir de manera inmediata a la aplicación. Entre estos módulos se encuentran algunos como el tratamiento de sesiones de usuario o el enlace con la *API*^[A5] de Google, reduciendo a una línea de código la implementación necesaria en la aplicación para el funcionamiento de estos.

Por tanto, el proyecto se basará en *Polymer* para la creación de la aplicación web.

Dentro de la propia aplicación, esta se dividirá en 4 grandes zonas:

- **Inicio:** Página principal con información sobre la aplicación y la empresa, además de otros datos informativos y acceso a la búsqueda de viajes.
- **Viajes:** Zona de búsqueda, creación y visualización de viajes y todo lo relacionado con estos.
- **Perfil:** Zona de información de usuario, ya visualización o edición del perfil del propio usuario o la consulta de perfiles de otro usuario.
- **Tracking de paquetes:** Información relativa a paquetes y su estado.

La aplicación se ha creado de forma que se descargue completamente al entrar en ella por primera vez el usuario y que no tenga que ser recargada al cambiar de página, con ayuda de la tecnología *AJAX*^[A1]. De esta forma la navegación por la página web es completamente fluida.

También se ha optado por realizar casi todo el trabajo de procesamiento en el lado del cliente, haciendo uso de *JavaScript*^[5.1.1.1]. Esto permite que el usuario tenga una mayor velocidad de respuesta en sus acciones y libera carga al servidor, además de reducir los datos transportados por red.

5.2.4 Sistemas de sesiones^{[B8] [B9]}

El sistema de sesiones creado se divide en dos partes, aquella referente al trato de sesiones por parte del servidor, y aquella que guarda datos localmente en el navegador del usuario. Vamos a proceder a explicar cada una de las dos partes.

5.2.4.1 Sesiones en la parte de servidor

Para las sesiones usaremos un plugin de *express* llamado “*express-session*” que crea la sesión en sí, guardando los datos de sesión en el servidor.

Este plugin creará automáticamente una cookie que compartirá con el que guardará los datos que deben compartir ambos. En el caso de nuestra aplicación únicamente guardará el *ID*^[A7] de sesión.

El plugin permite guardar los datos de sesión de diversas formas (por ejemplo almacenamiento de las sesiones basadas en distintas bases de datos) y en nuestro caso usaremos un sistema de guardado de datos por fichero, basado en una estructura *JSON*^[A15], de modo que cada sesión será un fichero separado que contendrá su información.

Se decidió por el uso de este método de almacenamiento para separar los datos persistentes guardados por la aplicación de los datos volátiles como son los de sesión. De esta forma se ahorra espacio en la base de datos principal (y ocupa menos espacio que crear una base de datos secundaria solo para guardar las sesiones) y permite mantener la modularización que se desea mantener en general en todo el proyecto y en particular en la aplicación.

5.2.4.2 Sesiones en la parte de cliente

Al iniciar sesión el usuario en la aplicación el servidor crea una cookie la cual envía al cliente, almacenándose ésta en el navegador usado por el usuario. Esta cookie contendrá en nuestro caso, como se ha comentado anteriormente, el *ID*^[A7] de la sesión actual.

Al iniciarse entonces la sesión se activa el módulo de *Polymer* “*ironlocal-storage*” para, mientras esté activa la sesión, guardar datos localmente. Estos datos contienen la información necesaria para mejorar la fluidez de navegación del usuario únicamente, de modo que no tenga que solicitar continuamente los mismos datos al servidor, ahorrando tiempo y tráfico de red tanto al usuario como al servidor.

5.2.5 Sistema de notificaciones

En la aplicación se ha creado un sistema de notificaciones con la intención de que, al crearse nuevas notificaciones el usuario esté al tanto de estas sin tener que volver a recargar la página en la que se encuentra. Para lograr esto existen tres formas posibles:

- **Pushing:** Cada vez que se crea una notificación, el servidor comprueba si el usuario al que le corresponde esa notificación tiene una sesión activa y le envía la nueva notificación. Esta opción está a partir de *HTML5*^[A11] y requiere de una gran complejidad de implementación al tratarse de un proceso asíncrono en el que el servidor manda datos al cliente sin que él haya mandado una solicitud para ello.
- **Polling/pulling:** Otra opción es que el cliente pregunte al servidor por las nuevas notificaciones cada cierto tiempo, haciendo que el servidor solo tenga que responder a esta pregunta. Aunque consume más tráfico de red.
- **Long polling:** Sistema que mejora la opción anterior, haciendo parecer más a la primera opción sin los inconvenientes de esta. El cliente realiza una solicitud de nuevas notificaciones con un *timeout*^[T9] de respuesta muy amplio (varios minutos). Si existen nuevas notificaciones el servidor enviará la respuesta inmediatamente. En caso contrario la solicitud del cliente queda pendiente durante ese tiempo. Si se genera una notificación en ese periodo, el servidor responderá al cliente con esta. Si caduca el *timeout*^[T9] o responde el servidor el cliente genera una nueva petición de nuevas notificaciones para continuar el ciclo.

La tercera opción sería la óptima a implementar, pero requiere un gran esfuerzo de configuración e implementación, por lo que se ha descartado temporalmente, dejando abierta la posibilidad de cambio a esta opción en un futuro como mejora al sistema de notificaciones.

En nuestra aplicación entonces se optó por la segunda opción, que necesita mucha menos configuración y aligera las tareas del servidor.

Por tanto el proceso que se sigue es el siguiente:

- El usuario inicia sesión y solicita al servidor todas las notificaciones relacionadas con él.
- El usuario pregunta cada intervalo de tiempo programado (en nuestro caso de un minuto) si existen nuevas notificaciones suyas. En caso afirmativo el servidor manda esas nuevas notificaciones.

5.2.6 Conexión con la API de Google^{[B14] [B15] [B16]}

Para este proyecto se ha necesitado el uso de la API^[A5] que proporciona Google con respecto a mapas y direcciones. Esto nos ha permitido automatizar bloques de la aplicación y mejorar la experiencia del usuario a la hora de tratar con la aplicación, haciéndola más intuitiva y clara.

Para tratar con la API^[A5] de Google tenemos dos posibilidades. Podemos tratar con ella directamente o bien podemos hacer uso de los módulos ya creados por *Polymer* para el trato mucho más rápido y claro con esta. En esta aplicación se ha decantado por la segunda opción.

El módulo usado de *Polymer* es el “*google-map*” y permite tratar con los tres bloques necesarios para la aplicación que explicaremos a continuación.

5.2.6.1 Google Directions API

Esta API^[A5] nos permite obtener información acerca de la ruta entre dos direcciones, usando el código que se muestra a continuación:

```
<google-map
  map="{{map}}"
  libraries="places">
</google-map>

<google-map-directions map="{{map}}"
  start-address="San Francisco"
  end-address="Mountain View"
  travel-mode="TRANSIT"
  waypoints="[{{location: 'Palo Alto'}}, {{location: 'San
Mateo'}}]">
</google-map-directions>
```

La primera línea de código es la encargada de crear un mapa de Google.

Las siguientes líneas conforman el módulo que trata con la **Google Directions API**. Para ello toman el mapa creado antes, busca la ruta más corta entre la dirección de inicio (*start-address*) y la de destino (*end-address*). Además se pueden añadir otros parámetros a la búsqueda como el transporte a usar para la ruta (avión, coche, caminando, etc) y puntos intermedios por los que debe pasar la ruta. Finalmente, devuelve esta ruta con toda su información, como duración de esta, sitios por los que pasa (cada sitio incluye información de latitud, longitud, nombre de pueblo o ciudad, provincia, país, etc.), kilómetros totales y kilómetros entre cada localización por la que pasa, etc.

Estos datos obtenidos de la ruta nos permitirán determinar los kilómetros aproximados del viaje que va a realizar un usuario y, a partir de estos, estimar un precio a pagar por cada pasajero que se una al viaje o por cada paquete que se quiera llevar, todo de forma automática.

Además, permite saber, conociendo la hora de salida del viaje, las horas aproximadas de llegada a cada punto intermedio del viaje, duración total del viaje y fecha de llegada. La aplicación además permite añadir tiempos de retardo en las localizaciones en las que el conductor realiza paradas intermedias en el viaje, por lo que la fecha y hora de llegada final del viaje también se calculan teniendo esto en cuenta.

5.2.6.2 Google Maps API^[B14]

Haciendo uso del código anterior, podemos mostrar la ruta seleccionada dentro del mapa, de modo que el usuario puede ver de forma gráfica y clara el trayecto que realizará un viaje en concreto al ver su información detallada.

Además, contamos con el siguiente código:

```
<google-map
  latitude="37.77493"
  longitude="-122.41942"
  fit-to-markers>
  <google-map-marker
    latitude="37.779"
    longitude="-122.382"
    draggable="true"
    title="Go Giants!">
  </google-map-marker>
  <google-map-marker
    latitude="37.77"
    longitude="-122.311">
  </google-map-marker>
</google-map>
```

Con esto pueden realizarse marcas en el mapa en la latitud y longitud deseada. Ayudándonos de este módulo podemos mostrar dentro de la ruta de un viaje los puntos intermedios de éste, mostrando todo el recorrido y paradas de forma mucho más clara.

Además, esta función ha sido utilizada también a la hora de crear viajes. Al elegir una ruta, esta puede tener distintos puntos de origen y destino. Para indicar cuáles son, el usuario puede, o bien escribirlos directamente en el campo destinado a ello del formulario de creación, o bien hacer click en las marcas que se crean dentro del mapa con las distintas posibilidades de origen y destino, rellenándose automáticamente estos campos en el formulario. Esto facilita la creación de viajes en gran manera ya que ayuda al usuario a decidir una localización de forma visual.

5.2.6.3 Google Geocoding API^[B15]

Para esta parte se ha realizado un módulo propio de *Polymer*, ya que el existente que trataba con Google no aportaba los datos que se necesitaban en la aplicación. Este módulo recoge la información de la API^[A5] de Google acerca de la latitud y longitud de una localización y añade una marca al mapa con esta localización.

Esto permite que, con un nombre de localización, el módulo encuentre su posición en el mapa y la marque. El código para usar este módulo es el siguiente:

```
<google-map-address-marker
  map="{{map}}"
  address="{{[address(item)]}}"
  identifier="{{[item.city]]}}"
  on-selected="selectMarkOrig">
</google-map-address-marker>
```

Con este código se crea para cada ciudad una marca, por lo que debe repetirse para cada una de las ciudades que se quieran marcar en un mapa.

5.3 Diseño final de la interfaz

En este apartado veremos los diferentes elementos y módulos de la interfaz descritos en el apartado de “*Diseño de Interfaz*” con más detalle, analizando cada elemento y su funcionalidad dentro del programa.

5.3.1 Área principal

Esta área será la estructura básica que tendrá toda la aplicación web, y consta de tres partes diferenciadas, que veremos en mayor detalle a continuación.

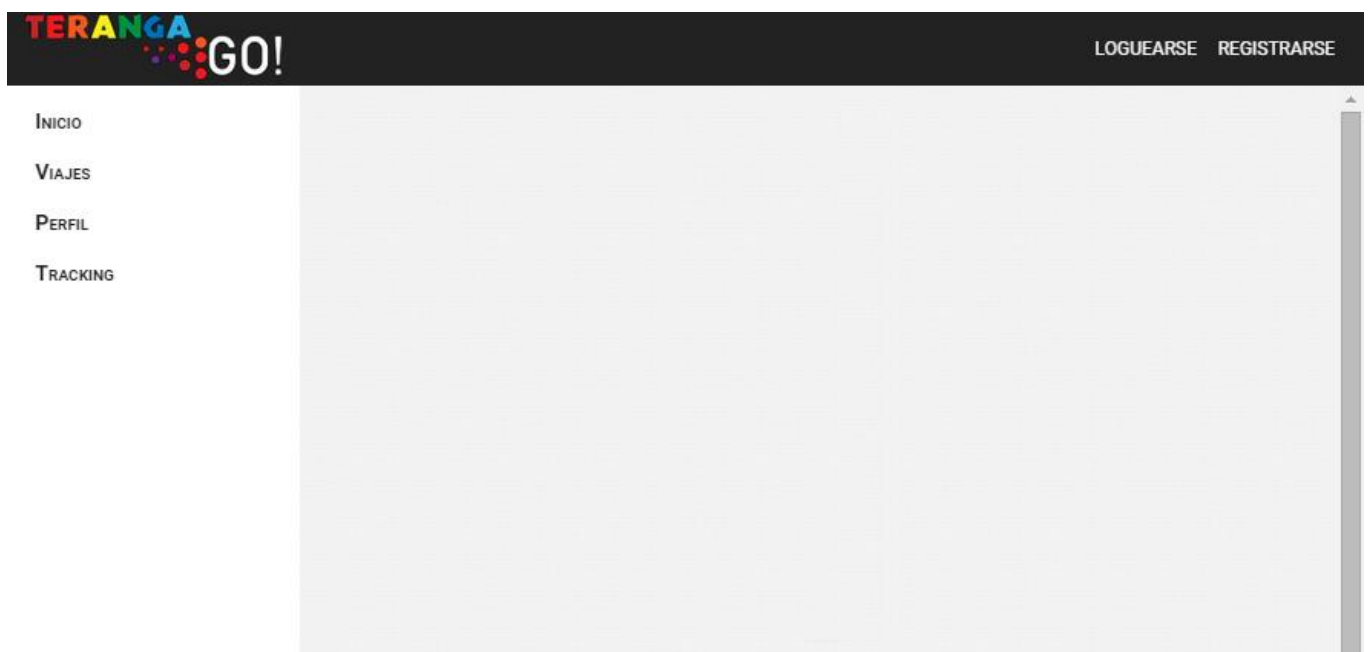


Imagen 83 - Interfaz – Área principal

5.3.1.1 Barra superior

En la zona superior se encuentra una barra con el nombre de la aplicación, así como de botones asociados a la sesión del usuario.



Imagen 84 - Interfaz – Barra superior

En caso de no haber una sesión iniciada se muestran dos botones:

- **Iniciar sesión:** Muestra un pop-up para permitir a un usuario ya registrado introducir sus datos únicos de correo y contraseña para iniciarse dentro de la aplicación y poder acceder a sus datos.
- **Registrarse:** Permite a cualquier persona crearse una cuenta, llevándola a la página de creación de nuevo usuario.

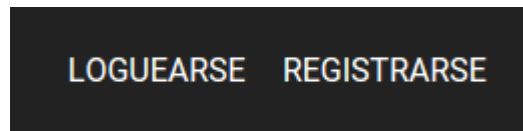


Imagen 85 - Interfaz – Barra superior sin sesión iniciada

Si ya se encuentra un usuario con la sesión iniciada, los botones anteriores se sustituyen por los mostrados en la imagen siguiente:



Imagen 86 - Interfaz – Barra superior con sesión iniciada

Estos dos botones cumplen las siguientes funciones:

- **Notificaciones:** Abre un menú en el cual se muestran listadas las notificaciones de un usuario, tanto leídas como sin leer. Cada una de estas notificaciones son un enlace a información más detallada acerca de la notificación.
- **Botón de perfil:** Abre un menú el cual muestra dos opciones:
 - **Ver perfil:** Redirige a la página de perfil del usuario.
 - **Cerrar sesión:** Termina la sesión del usuario.

5.3.1.2 Menú lateral

Se sitúa en el lateral izquierdo de todas las páginas de la aplicación. Muestra una lista con enlaces a las páginas más representativas de la aplicación:

- **Inicio:** Te redirige a la página principal de la aplicación.
- **Perfil:** Te enlaza a la página de tu perfil en caso de tener una sesión iniciada o a la página de registro en caso contrario.
- **Viajes:** Te permite buscar entre los viajes existentes.
- **Tracking:** Te lleva a una página para controlar la localización y estado de tus paquetes.

5.3.1.3 Área central

En este área se visualizarán las distintas páginas de la aplicación.

5.3.2 Zona de perfil / registro

Se puede acceder a estas páginas tanto desde el menú lateral (“*Perfil*”) o desde los botones de la barra superior (“*registrarse*” si no hay sesión iniciada y “*Ver perfil*” en caso contrario)

5.3.2.1 Registro

Cuando un usuario no ha iniciado sesión, al entrar en la página de perfil o al hacer click en el botón de registrarse, es conducido a la página de registro, en donde podrá introducir sus datos para convertirse en usuario registrado de la página.

Imagen 87 - Interfaz – Página de registro

Para poder realizar el registro debe rellenar de forma obligatoria los siguientes campos:

- **Nombre y apellidos:** Cadenas de caracteres con información acerca de su nombre y apellidos reales.
- **E-mail:** Correo electrónico real del usuario (Se realizará una validación del correo antes de permitir al usuario terminar el registro)
- **Contraseña y verificación:** Cadena alfanumérica de al menos 8 caracteres que tendrá que repetir dos veces.

Además de estos campos, el usuario podrá introducir información opcional acerca de los siguientes campos:

- **Imagen de perfil:** Imagen elegida por el usuario.
- **Teléfono:** Cadena de números indicando el teléfono de contacto del usuario.
- **Fecha de nacimiento:** Para establecer la edad del usuario y mostrarla a otros.
- **Conductor:** Indica si el usuario es conductor o no.

En el lateral derecho, además, se encuentran los siguientes botones:

- **Editar perfil:** Accede a la página de editar perfil explicada más adelante.
- **Crear viaje:** Accede a la página de creación de viajes explicada más adelante.

5.3.2.2 Inicio de sesión

Para poder entrar como usuario registrado es necesario rellenar el siguiente formulario de logueo, que aparece como un pop-up sobre la página.

El formulario de inicio de sesión se presenta como un pop-up. A la izquierda hay una línea vertical que separa el formulario del resto de la interfaz. El formulario contiene dos campos de texto: 'Email' y 'Contraseña'. Debajo de estos campos hay un botón rectangular de color naranja con el texto 'INICIAR' en blanco.

Imagen 88 - Interfaz – Formulario de logueo

En este contenedor se deben introducir los siguientes campos de texto:

- **E-mail:** Correo con el que se ha registrado el usuario.
- **Contraseña:** Cadena alfanumérica de mínimo 8 caracteres que indicó en el registro.

Se deberán introducir obligatoriamente ambos campos para poder continuar. Se realizará una comprobación con la base de datos y, en caso de existir un usuario con ese correo y contraseña, iniciará sesión en la aplicación. En caso contrario se indicará así en el pop-up.

5.3.2.3 Perfil propio

Una vez registrados podemos consultar nuestro perfil:



Imagen 89 - Interfaz – Perfil propio

El perfil se divide en tres zonas separadas por delimitadores en forma de líneas horizontales. A continuación explicaremos cada zona de estas:

5.3.2.3.1 Información de usuario

En esta zona se muestra información acerca del propio usuario:

- Nombre y apellidos.
- Edad (En caso de haber sido introducida una fecha de nacimiento)
- Imagen de perfil (En caso de haber sido introducida por el usuario, en caso contrario, se mostrará una imagen por defecto)

Además, se muestra, en caso de haber realizado algún viaje, la puntuación como conductor media que ha obtenido. Esto se mostrará de forma gráfica.



Imagen 90 - Interfaz - Puntuación

5.3.2.3.2 Desplegables

En la zona central se muestran una serie de desplegables con diversas informaciones útiles para el usuario:

- **Viajes:** Muestra los viajes pendientes del usuario, tanto los propios como a los que ya ha sido aceptado.
- **Solicitudes:** Muestra las solicitudes pendientes del usuario, tanto las que él ha creado para pedir viajar en un viaje de otro como las que otros usuarios le han realizado para viajar en un viaje creado por él.
- **Paquetes:** Muestra aquellos paquetes que tiene el usuario, así como su estado.
- **Usuarios favoritos:** Muestra información de todos los usuarios que el usuario ha marcado como favoritos, así como un enlace al perfil de cada uno.

La información que se muestra dentro de los desplegables es una lista de módulos que se pueden ver en el *apartado 5.3.5*.

Todos los desplegables cuentan con el mismo estilo y funcionamiento, aunque algunos cuentan con botones extras:

- **Añadir:** Permite crear un elemento del tipo que contiene el desplegable. Disponible en los desplegables de paquetes y viajes.
- **Recargar:** Realiza una petición a la base de datos solicitando cambios en los elementos que contiene el desplegable. Todos los desplegables tienen este botón.

5.3.2.3.3 Opiniones

En caso de haber realizado algún viaje, el usuario tendrá un tercer apartado en la parte inferior de su perfil, el cual mostrará las opiniones y puntuaciones que ha recibido como conductor.

Estas opiniones estarán ordenadas cronológicamente, mostrando primero las más recientes y se mostrarán hasta un máximo de 10 opiniones.

5.3.2.4 Editar perfil

En la zona de editar perfil el usuario puede modificar algunos de sus datos, como veremos a continuación. En esta página se muestran dos zonas separadas por delimitadores.

5.3.2.4.1 Información de usuario

Imagen 91 - Interfaz – Editar perfil – Información de usuario

Esta zona permite al usuario modificar los siguientes datos:

- **Teléfono:** Número de teléfono de contacto.
- **Conductor:** Si es o no conductor.
- **Imagen de perfil:** Imagen que le representa como usuario.
- **Contraseña:** cadena alfanumérica de al menos 8 caracteres necesaria para iniciar sesión.

En el lateral derecho se muestran dos botones:

- **Guardar cambios:** Permite actualizar los datos con la nueva información introducida por el usuario.
- **Cancelar:** Vuelve a la página de perfil sin realizar ningún cambio.

5.3.2.4.2 Coches

En esta zona se muestra información básica sobre los coches que ha indicado que posee el usuario. Cada coche contará además con un enlace que mostrará un pop-up con su información detallada, así como la posibilidad de modificar sus datos.

Además se podrá añadir un nuevo coche, haciendo click en el último módulo de la lista, que abrirá un pop-up con un formulario a rellenar por el usuario con la información acerca del nuevo coche.

5.3.2.5 Perfil de otro usuario



Imagen 92 - Interfaz – Perfil de otro usuario

Al visitar el perfil de otro usuario se mostrará una página dividida en tres zonas, las cuales explicaremos con más detalle a continuación.

5.3.2.5.1 Información general

Esta zona central muestra información general acerca del usuario:

- **Nombre y apellidos**
- **Imagen de perfil**
- **Edad** (en caso de que el usuario la haya especificado)

En caso de que el usuario visitado haya realizado algún viaje, debajo de su imagen de perfil se mostrará gráficamente la puntuación media obtenida de sus viajes, al igual que en el perfil propio.

Además de esto, en la parte derecha aparecerá un botón para añadir el usuario como favorito en caso de no tenerlo como favorito, o eliminarlo de favoritos en caso contrario.

5.3.2.5.2 Comentarios

En la zona central inferior, en caso de que el usuario visitado haya realizado algún viaje, el usuario tendrá un tercer apartado en la parte inferior de su perfil, el cual mostrará las opiniones y puntuaciones que ha recibido como conductor.


Estas opiniones estarán ordenadas cronológicamente, mostrando primero las más recientes y se mostrarán hasta un máximo de 10 opiniones.

5.3.2.5.3 Viajes programados

En caso de que el usuario sea conductor, aparecerá esta tercera zona en el lateral derecho. Esta zona contendrá un desplegable en el cual se mostrarán, en caso de haberlos, los viajes próximos creados por el usuario visitado.

5.3.3 Viajes

5.3.3.1 Creación de viajes




Ruta
Ruta1

De: Granada

A: Portos

Fecha Hora



Coche





Asientos libres

Precio

Asiento x 13€

Paquete x 10€

TRAYECTO COMPLETO

Granada		<input type="text"/>		Salida	<div>10 Jun</div>	<div>10:30</div>
Motril		<input type="text"/>	Retardo	<input type="text" value="--:--"/>	Salida	<div>10 Jun</div> <div>10:30</div>
Gibraltar		<input type="text"/>	Retardo	<input type="text" value="--:--"/>	Salida	<div>10 Jun</div> <div>10:30</div>
Portos		<input type="text"/>				

GUARDAR

CANCELAR

Imagen 93 - Interfaz – Creación de viajes

Accediendo desde el perfil se encuentra la página de creación de viaje. Esta página se encuentra dividida en dos zonas para especificar todos los datos del viaje.

5.3.3.1.1 Detalles del viaje

En esta parte el usuario debe indicar los siguientes datos de forma obligatoria:

- Ruta a utilizar.
- Localización de salida de entre las posibles de la ruta anterior y la dirección desde donde se saldrá.
- Localización de destino de entre las posibles de la ruta anterior y la dirección donde se terminará el viaje.
- Fecha de salida (día y hora)
- Preferencias de llevar mascotas o paquetes de otros pasajeros.
- Coche con el que se va a realizar el viaje.
- Asientos libres que podrán coger otros usuarios.
- Precio por persona del viaje (se escogerá sobre una base el multiplicador, permitiendo variar ligeramente el precio establecido por la aplicación para ese viaje según los kilómetros a realizar).

De forma opcional se podrá añadir un comentario acerca del viaje.

5.3.3.1.2 Paradas intermedias

Se mostrará una lista con las localizaciones intermedias que tiene la ruta seleccionada. Para cada localización, el usuario deberá indicar si va a realizar una parada (y por tanto permitir bajarse y subirse nuevos pasajeros) o no.

En caso de realizar una parada, el usuario deberá indicar la dirección exacta en la que realizará la parada y si va a tener tiempo de retardo antes de volver a retomar el viaje.

5.3.3.2 Búsqueda de viajes



The image shows a travel search interface. On the left, a map of Africa displays a blue route from Dakar to Granada. A callout box on the map indicates a travel time of 44 hours and a distance of 3,511 km. On the right, the search form is titled "ENCUENTRA EL VIAJE QUE NECESITAS". It includes a "De:" field with "Granada" and an "A:" field labeled "Destino" with a red underline and the text "Seleccione una opción de la lista". Below this, the "CUANDO LO NECESITAS" section has "Desde:" and "Hasta:" date pickers set to "29/07/2015" and "10/12/2015" respectively. There are two groups of icons: the first group contains a paw print, a grey paw print, and a paw print with a red X; the second group contains a box, a grey box, and a box with a red X. At the bottom is a large orange button labeled "BUSCAR".

Imagen 94 - Interfaz – Búsqueda de viajes

Esta página permite realizar una búsqueda de los viajes existentes entre un origen y destino determinados. Para ello, el usuario debe introducir los siguientes datos de búsqueda:

- Origen desde el que quiere salir.
- Destino al que quiere llegar.
- Intervalo de fechas en las que quiere viajar (día, mes y año).
- Preferencias en cuanto a viajar junto a mascotas o paquetes.

Una vez introducidos los datos, se mostrará una lista de aquellos viajes que coincidan con la búsqueda. Los viajes no necesariamente tienen que empezar en el origen y acabar en el destino indicados por el usuario, pueden tener paradas intermedias con estas localizaciones.

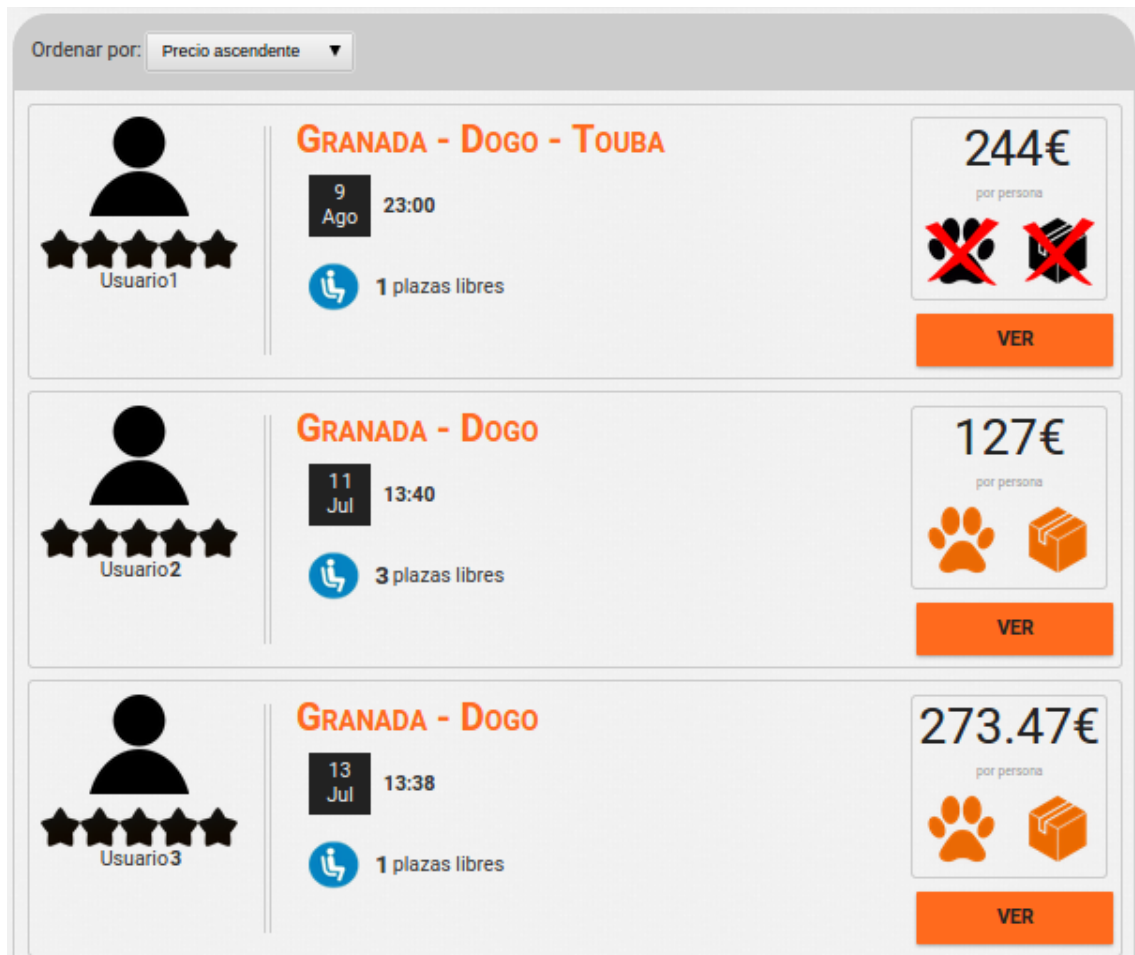


Imagen 95 - Interfaz – Lista de viajes

Cada viaje mostrará los siguientes detalles:

- Conductor que ha creado el viaje (nombre, foto de perfil en caso de tenerla y puntuación media obtenida por otros usuarios).
- Origen y destino del viaje y paradas intermedias que ha pedido el usuario de la búsqueda.
- Fecha y hora de salida del origen indicado por el usuario en su búsqueda.
- Plazas libres para el trayecto que va a realizar el usuario.
- Precio por persona del trayecto que va a realizar el usuario.
- Enlace a la página con información detallada del viaje.

Además, la lista de los viajes podrá ser ordenada según distintos criterios:

- Precio ascendente.
- Precio descendente.
- Fecha y hora.
- Puntuación media de los conductores de los viajes.

5.3.3.3 Información de viaje

The screenshot displays the travel information for the route **GRANADA - TOUBA**. On the left, a vertical list of stops is shown: **GRANADA** (9 Ago, 23:00, 2 seats), **GIBRALTAR**, **DOGO** (1 Jul, 14:50, 1 seat), and **TOUBA** (10 Ago, 09:30, 3 seats). The main section for the route includes the title **GRANADA - TOUBA**, the departure date **Fecha de salida 9 Ago**, and the departure time **Hora de salida 23:00**. The price is listed as **8€ por persona**, with icons indicating that pets and bicycles are not allowed. Below this, the **CONDUCTOR** section shows a profile for **Usuario1** (24 años, usuario1@mail.com, 1111111111) with a 4-star rating. The **COCHE** section shows a **Model1** (M1XS) with **1 asientos libres**. At the bottom, the **COMENTARIO DEL CONDUCTOR** section contains a placeholder text: "El conductor comenta que.... blablabla Comentario blablabla Comentario blablabla Comentario blablabla Comentario blablabla Comentario blablabla Comentario blablabla Comentario blablabla Comentario blablabla Comentario blablabla.... y eso es todo".

Imagen 96 - Interfaz – Información de viaje

Al visitar la página de un viaje, se puede ver toda la información detallada de este, dividida en dos zonas, las cuales pasaremos a explicar con más detalle a continuación.

5.3.3.3.1 Información del viaje

Esta zona central cuenta con la información del viaje, separada en los siguientes puntos:

- Localización de origen y destino del viaje.
- Fecha y hora de salida.
- Precio por persona
- Preferencias de viaje con respecto a mascotas y paquetes del conductor.
- Información del conductor y creador del viaje
 - Nombre
 - Imagen de perfil en caso de tenerla
 - Edad en caso de haberla indicado
 - Correo de contacto
- Información sobre el coche en el que se hará el viaje
 - Modelo del coche
 - Matrícula
 - Asientos libres disponibles para realizar el viaje completo.
- Comentario del conductor acerca del viaje
- Enlace para solicitar viajar en ese viaje como pasajero o mandar algún paquete.

5.3.3.3.2 Localizaciones intermedias

Esta lista muestra todas las localizaciones intermedias del viaje. Indicando si va a realizarse o no una parada en ellas. En caso de realizarse una parada, además se muestran los siguientes datos de la parada:

- Fecha y hora de salida (Contando el retardo que haya indicado el conductor)
- Asientos disponibles al retomar el viaje en esa parada.

5.3.3.4 Solicitud de viaje

PEDIR

Salida: Granada ▼ 25/Nov 12:40
Dirección: aaa

Llegada: Dogo (313€) ▼ 25/Nov 12:40
Dirección: zzz

Comentario

Asiento ☒ +127€
Paquetes +186€ (2x 93€)

Paquete1 Grande X
Es un paquete del usuario 3

Paquete2 Muy Grande X
Este es otro paquete del usuario 3

AÑADIR PAQUETE CREAM PAQUETE

Precio final 313€ **SOLICITAR**

Imagen 97 - Interfaz – Solicitud de viaje

Al acceder en la parte inferior de un viaje a su solicitud se abre un pop-up ara rellenar los datos necesarios. Estos son:

- Origen del que quieres salir de los disponibles del viaje
- Destino en el que quieres bajarte entre los disponibles del viaje.
- Comentario acerca de tu solicitud del viaje que recibirá el conductor.
- Si se quiere viajar o no como pasajero.

En caso de que el viaje acepte paquetes, se permitirá escoger si se quieren añadir paquetes al viaje. Para ello se deberá escoger un paquete de entre los que ya posee el usuario o bien crear uno desde el principio.

Una vez escogido el paquete a llevar este se añade a la lista de paquetes a llevar y se actualiza el precio total del viaje.

Se podrán eliminar y añadir los paquetes que se quieran de la lista antes de confirmar la solicitud.

5.3.4 Paquetes

La aplicación permite crear paquetes para mandarlos en un viaje y llevar un control de ellos mientras están de camino.

5.3.4.1 Detalle de paquete



ID

Introduce el identificador (ID) del paquete

El identificador debe estar compuesto únicamente por números

BUSCAR

Imagen 98 - Interfaz – Búsqueda de paquete

Accediendo desde el menú lateral de la aplicación se puede acceder a la página de información de paquetes. En primer lugar se mostrará una página para introducir el ID del paquete a mostrar. Debe introducirse un ID de paquete válido para mostrar alguna información.

El ID^[A7] de paquete es un número.

Una vez introducido el paquete se muestra una página con información acerca de este. Esta información se divide en dos zonas que pasaremos a explicar a continuación.



Imagen 99 - Interfaz – Información paquete

5.3.4.1.1 Información de paquete

En esta página se muestran los datos del paquete facilitados por el usuario que lo creó. Estos son:

- Nombre del paquete.
- Identificador del paquete.
- Tamaño del paquete.
- Comentario del creador del paquete sobre este.

En esta zona se permite además eliminar el paquete con el botón con el mismo nombre.

5.3.4.1.2 Localización del paquete


Esta zona muestra información acerca del estado y localización del paquete.

En primer lugar muestra el estado del paquete, para indicar si está o no viajando. En caso de estar viajando se muestra la siguiente información:

- Fecha estimada de salida del paquete.
- Fecha estimada de llegada del paquete.
- Última localización que indicó el conductor del viaje en el que viaja el paquete.
- Indicador gráfico de la localización del paquete dentro de la ruta que debe recorrer.

5.3.4.2 Creación de paquete

Desde el perfil o el formulario de solicitud de viaje se puede crear un nuevo paquete en el que se indica datos generales del paquete.



Formulario de creación de paquete:

NUEVO PAQUETE

Nombre:

Tamaño: 4

Descripción: 0/200

Usuario receptor y/o DNI, NIF o NIE del receptor:

CREAR

CANCELAR

Imagen 100 - Interfaz – Creación de paquete

Los datos a indicar para la creación de un paquete son los siguientes:

- Nombre del paquete.
- Tamaño aproximado del paquete.
- Comentario acerca del paquete.

Una vez indicados los datos mostrados anteriormente puede crearse el paquete desde el botón correspondiente en la parte inferior. Esto además llevará a la página de información del nuevo paquete creado y generará un ID^[A7] para este paquete.

5.3.5 Módulos

Para mostrar información básica de diversos elementos de la aplicación se han creado módulos para mostrar esta información de forma clara y entendible.

5.3.5.1 Viaje

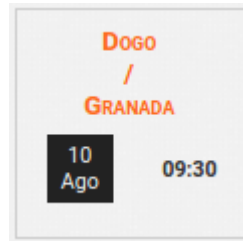


Imagen 101 - Interfaz – Módulo viaje

Muestra información acerca de un viaje en concreto:

- Localización de salida.
- Localización de destino.
- Nombre del conductor.
- Fecha de salida.
- Hora de salida.

Al hacer click en este módulo, este lleva a la página con la información detallada acerca del viaje.

5.3.5.2 Solicitud



Imagen 102 - Interfaz – Módulo solicitud

Muestra información básica acerca de una solicitud de viaje:

- Localización de origen del viaje solicitado.
- Localización de destino del viaje solicitado.

Si es una solicitud de viaje que el usuario ha realizado a otro, se muestra además la

siguiente información:

- Estado de la solicitud, que puede variar entre:
 - Pendiente: El conductor del viaje aún no ha respondido a la solicitud.
 - Aceptada: El conductor del viaje ha aceptado tu solicitud.
 - Rechazada: El conductor del viaje ha rechazado tu solicitud.
 - Expirada: La fecha de salida del viaje ya ha pasado y el conductor no ha respondido a tu solicitud.
- Fecha de salida del viaje.
- Hora de salida del viaje.

Al hacer click en este módulo te lleva a la página de información detallada del viaje que se ha hecho la solicitud.

En cambio, en caso de tratarse de una solicitud que otros han realizado al usuario se mostrará la siguiente información:

- Número de asientos que quiere ocupar el usuario (0 o 1).
- Número de paquetes que el usuario quiere enviar en el viaje (Mayor o igual que 0).
- Enlace a la información detallada de la solicitud.

5.3.5.3 Coche

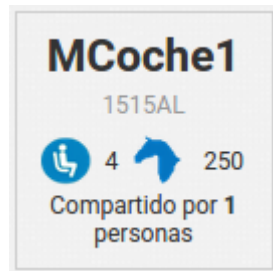


Imagen 103 - Interfaz – Módulo coche

Muestra información acerca de un coche:

- Modelo del coche.
- Matrícula.
- Asientos máximos que tiene (Incluyendo el conductor)
- Caballos de potencia.
- Número de usuarios que han indicado que usan ese mismo coche.

5.3.5.4 Paquete



Imagen 104 - Interfaz – Módulo paquete

Muestra información básica sobre un paquete:

- Nombre del paquete.
- ID del paquete.
- Tamaño del paquete.
- Estado en el que se encuentra actualmente el paquete.

Al hacer click en este módulo te lleva a la página con la información detallada del paquete.

5.3.5.5 Usuario favorito



Imagen 105 - Interfaz – Módulo usuario favorito

Este módulo muestra información básica sobre un usuario el cual ha sido añadido como favorito:

- Imagen de perfil en caso de haber (Si no existe se mostrará una imagen por defecto).
- Puntuación media del usuario realizada por otros usuarios acerca de sus viajes en caso de haber realizado alguno.
- Nombre del usuario.
- Número de viajes activos actualmente del usuario.

Al hacer click sobre este módulo te enlaza a la página de perfil del usuario en cuestión.

5.3.5.6 Opinión / Comentario

Muestra un comentario realizado por un usuario:

- Imagen de perfil en caso de existir del usuario que realiza el comentario.
 - Esta imagen tendrá un enlace al perfil de este usuario.
- Puntuación que da el usuario al conductor que creó el viaje, valorada de 0 a 10.
- Puntuación que da el usuario al coche en el que se realizó el viaje, valorada de 0 a 10.
- Comentario de texto acerca del viaje y la puntuación dada.

6. Conclusiones

6.1 Objetivos alcanzados

En este apartado retomaremos los objetivos planteados al comienzo de la memoria para analizar cuales se han cumplido, cuales no y por qué.

Objetivo 1	<i>Identificación única de usuario</i>
Tipo	Obligatorio
Conseguido	Si
Descripción	Obliga a un usuario a identificarse para poder interactuar con la aplicación

Objetivo 2	<i>Creación de viajes</i>
Tipo	Obligatorio
Conseguido	Si
Descripción	Permite a un usuario dar los datos acerca del viaje que va a realizar para permitir a otros usuarios unirse a él.

Objetivo 3	<i>Búsqueda de viajes según criterios</i>
Tipo	Obligatorio
Conseguido	Si
Descripción	Permite a un usuario encontrar los viajes que se adaptan a sus necesidades

Objetivo 4	<i>Solicitud de participación en viajes</i>
Tipo	Obligatorio
Conseguido	Si
Descripción	Permite a un usuario solicitar ir de pasajero en un viaje

Objetivo 5	<i>Solicitud de transporte de paquetes en viajes</i>
Tipo	Obligatorio
Conseguido	Si
Descripción	Permite a un usuario solicitar que un paquete de su propiedad vaya como equipaje en un viaje

Objetivo 6	<i>Sistema de trato de solicitudes</i>
Tipo	Opcional
Conseguido	Si
Descripción	Permite a un usuario decidir si aceptar o no las solicitudes realizadas por otros usuarios para su viaje.

Objetivo 7	<i>Sistema de puntuaciones de usuarios y viajes</i>
Tipo	Obligatorio
Conseguido	Si
Descripción	Permite a un usuario que ha ido de pasajero en un viaje puntuar el mismo y a su conductor, creando un sistema de puntuaciones de usuario que se mostrará públicamente.

Objetivo 8	<i>Control de localización de paquetes</i>
Tipo	Opcional
Conseguido	No
Motivo	Se ha creado el sistema para ver la localización del paquete, pero no el sistema para que el usuario indique en que localización del viaje se encuentra
Descripción	Permite a un usuario tener constancia de la localización aproximada de su paquete mientras este se encuentra en el trayecto de un viaje.

Objetivo 9	<i>Sistema de mensajería interna</i>
Tipo	Opcional
Conseguido	No
Motivo	No era necesaria para el correcto funcionamiento de la aplicación, por lo que se ha dejado el objetivo como pendiente para una implementación futura.
Descripción	Permite crear un sistema de mensajería por la cual los usuarios pueden de forma privada comunicarse entre sí dentro de la misma aplicación.

Objetivo 10	<i>Soporte de idiomas</i>
Tipo	Opcional
Conseguido	No
Motivo	No ha sido posible crear el soporte debido al ser imposible de integrar dentro de la planificación prevista, por lo que queda pendiente de añadir en una versión futura de la aplicación.
Descripción	Separar el texto mostrado en la aplicación, permitiendo una fácil transcripción de esta a otros idiomas.

6.2 Lecciones aprendidas

La elaboración de este proyecto me ha permitido aprender y mejorar en muchos campos del desarrollo software. Aunque ya conocía y manejaba el lenguaje de programación principal usado en el proyecto (*JavaScript*), he podido mejorar y conocerlo más a fondo.

Además he podido profundizar en la creación de páginas web con la herramienta *Polymer*, con gran auge e importancia actualmente, lo cual me servirá para futuros proyectos.

Este proyecto me ha permitido también emplear y mejorar mis habilidades de diseño gráfico, ya que ha hecho falta un estudio acerca de cómo diseñar la interfaz de usuario, cómo distribuir los elementos para ser lo más amigables posibles al usuario, etc.

Además, la complejidad y tamaño del proyecto a abarcar me han permitido desarrollar mi organización del tiempo, habilidades para crear una documentación clara, estudio de diferentes opciones a la hora de desarrollar el software, etc.

Una vez finalizado este proyecto creo haber logrado la experiencia necesaria para afrontar otro proyecto de iguales o incluso mayores magnitudes con un resultado mucho más afinado y depurado que antes de empezar con este.

6.3 Vías futuras

El programa contiene toda la funcionalidad requerida, pero aun así quedan aspectos por tratar dentro de él para seguir facilitando la tarea del usuario al tratar con esta aplicación. Varias funcionalidades que podrían ser añadidas en un futuro para la mejora del programa serían:

- Mejorar el sistema de notificaciones tal y como se explicó en el ***apartado 5.2.5***. Mejorando así la eficiencia y reduciendo tiempos de espera y tráfico de datos.
- Añadir facilidades para la transcripción de la aplicación web a otros idiomas. Actualmente la web solo se encuentra disponible en español, por lo que añadir otros idiomas puede acercar la aplicación a un mayor grupo de usuarios.
- Crear la aplicación móvil con los mismos datos de la aplicación web y enlazada con esta puede facilitar el uso de la aplicación. Para esto solo se necesitaría la programación en *Android*^[14] y/o *iOs*^[15], ya que toda la parte de interacción con la base de datos ya está creada.
- Expandir la aplicación al transporte entre España y África, ya que ahora se centra solo entre España y Senegal. Ya que España es uno de los puntos de unión entre Europa y África, ampliar el ámbito de acción de la aplicación, cubriendo un mercado no explotado del tráfico compartido de pasajeros y paquetes entre España y África sería muy beneficioso.

7. Bibliografía

7.1 Enlaces de interés

- [B1] **Blablacar:** www.blablacar.es
- [B2] **Blablacar app:** <https://play.google.com/store/apps/details?id=com.comuto&hl=es>
- [B3] **Amovens:** www.amovens.com
- [B4] **Amovens app:**
<https://play.google.com/store/apps/details?id=com.amovens.pruebandroid&hl=es>
- [B5] **RoarSharing:** www.roadsharing.com/es
- [B6] **Acento comunicación:** <http://www.acentocomunicacion.com/>
- [B7] **NodeJS:** <https://nodejs.org/>
- [B8] **Express:** www.expressjs.com/es/
- [B9] **Polymer:** <https://elements.polymer-project.org/>
- [B10] **CSS:** <http://www.w3schools.com/css/>
- [B11] **HTML:** <http://www.w3schools.com/html/>
- [B12] **Postman:** <https://www.getpostman.com/docs>
- [B13] **MySQL Workbench:** <https://www.mysql.com/products/workbench/>
- [B14] **API Maps Google:** <https://developers.google.com/maps/>
- [B15] **API Geocoding Google:**
<https://developers.google.com/maps/documentation/geocoding/>
- [B16] **API Directions Google:**
<https://developers.google.com/maps/documentation/directions/>
- [B17] **Información general:** <https://es.wikipedia.org/>
- [B18] **InfoJobs:** <https://www.infojobs.net/>
- [B19] **GPL License:** <http://www.gnu.org/licenses/gpl.html>
- [B20] **Polymer License:** <http://polymer.github.io/LICENSE.txt>
- [B21] **NodeJS Express License:** <http://opensource.org/licenses/mit-license.php>
- [B22] **Creación diagramas de secuencia:** <https://www.websequencediagrams.com/>

8. Anexos

8.1 Código fuente y ejecutable

El código del programa puede encontrarse en el siguiente enlace:

- **Repositorio GitHub**
 - <https://github.com/FdezI/Teranga.Go>

Puede verse todo el proceso de instalación en la página indicada anteriormente.

8.2 Glosario de términos

8.2.1 Términos^[B17]

A continuación se listan los términos específicos del dominio del problema con una breve definición de cada uno.

- [T1] **Economía colaborativa:** Sistema económico en el que se comparten bienes y servicios haciendo uso de la tecnología de la información actual.
- [T2] **web component:** Estandar de HTML5 que permite crear módulos de widgets o componentes independientes y reutilizables en cualquier web.
- [T3] **carsharing:** Sistema de compartir coche entre varias personas desconocidas que quieren realizar el mismo viaje.
- [T4] **Android:** Sistema operativo para móviles basado en Linux.
- [T5] **iOS:** Sistema operativo para móviles de la compañía Apple.
- [T6] **Red Social:** Medio de comunicación que permite establecer contacto entre personas por internet que comparten algún tipo de relación o interés común.
- [T7] **web responsiva:** Página web que adapta su contenido al tamaño de la pantalla
- [T8] **Interfaz de usuario (UI):** Es el medio con que el usuario puede comunicarse con nuestro programa.
- [T9] **TimeOut:** Tiempo de espera en una solicitud del cliente o usuario al otro tras el cual se asume que no ha habido respuesta y se cierra la conexión.

8.2.2 Acrónimos^[B17]

A continuación se listan los acrónimos específicos del dominio del problema con su correspondiente significado.

- [A1] **AJAX**: *Asynchronous JavaScript And XML* (“JavaScript asíncrono y XML”), tecnología ejecutada en el navegador para crear aplicaciones interactivas.
- [A2] **AJAJ**: *Asynchronous JavaScript And JSON* (“JavaScript asíncrono y JSON”), tecnología ejecutada en el navegador para crear aplicaciones interactivas.
- [A3] **API REST**: *Application Programming Interface for Representational State Transfer* (“Interfaz de programación de aplicaciones para la Transferencia de Estado Representacional”), es un estilo de arquitectura que utiliza URLs para obtener información del servidor.
- [A4] **TIC**: *Tecnologías de la Información y la Comunicación*.
- [A5] **API**: *Application Programming Interface* (“Interfaz de programación de aplicaciones”), es el conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- [A6] **DB**: *DataBase* (Base de datos) (También BD).
- [A7] **ID**: *Identificador* de un elemento.
- [A8] **UML**: *Unified Modeling Language* (“Lenguaje Unificado de Modelado”), es un lenguaje gráfico para visualizar un sistema.
- [A9] **UTC**: *Coordinated Universal Time* (“tiempo universal coordinado”), estándar de tiempo con el que se regula el tiempo en el mundo.
- [A10] **ISO**: *International Organization for Standardization* (“Organización Internacional de Normalización”), organismo encargado de desarrollar normas internacionales para conseguir la estandarización de elementos a nivel mundial.
- [A11] **HTML**: *HyperText Markup Language* (“lenguaje de marcas de hipertexto”), es un lenguaje de marcado para la elaboración de páginas web.
- [A12] **DOM**: *Document Object Model* (“Modelo de Objetos del Documento” o “Modelo en Objetos para la Representación de Documentos”), es una API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.
- [A13] **CSS**: *Cascading Style Sheets* (“Hoja de estilo en cascada”), es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.
- [A14] **XML**: *eXtensible Markup Language* (“lenguaje de marcas extensible”), es un lenguaje de marcas utilizado para almacenar datos en forma legible.
- [A15] **JSON**: *JavaScript Object Notation* (“Notación de objetos en JavaScript”), formato ligero y simple utilizado para almacenar datos de forma legible.