



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍA INFORMÁTICA DE SEGOVIA

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

TripMerge

Aplicación web para planificar viajes entre distintas personas

Alumno: Rubén Velasco Plaza

Tutor: Fernando Díaz Gómez

Agradecimientos

Quiero dedicar mi Trabajo Final de Grado a mi familia y a mis amigos que me han apoyado y ayudado en todo momento. Sin ellos no hubiera sido posible llegar hasta aquí. Gracias, muchas gracias.

También quiero agradecer a mi tutor Fernando Díaz Gómez por aceptar el desarrollo de este proyecto y por su ayuda en todo momento.

Resumen

El objetivo principal de este Trabajo Final de Grado es el ofrecer la facilidad para gestionar un viaje individual o con varias personas. De esta forma, se crea *TripMerge*, una aplicación web que contiene varios apartados para almacenar equipaje, gastos y actividades que conlleva un viaje. Todo ello es compartido entre distintos usuarios para unificar el viaje y sea más accesible para las personas.

Palabras clave: Aplicación web, Gestión viajes, equipaje, gastos y actividades

Abstract

The main objective of this Final Degree Project is to offer the facility to manage an individual trip or a trip with several people. In this way, TripMerge is created, a web application that contains several sections to store luggage, expenses and activities involved in a trip. All this is shared among different users to unify the trip and make it more accessible to people.

Keywords: Web application, Travel, luggage, expense and activity management

Contenido

Índice de figuras	9
Índice de tablas	11
Capítulo 1: INTRODUCCIÓN	13
1.1 Organización del documento	13
1.2 Introducción	14
1.3 Motivación	14
1.4 Objetivos	15
1.5 Alcance	17
1.6 Entorno tecnológico	20
Capítulo 2: ESTADO DEL ARTE	25
2.1 TravelPerk España	25
2.2 Expensya	26
2.3 Google Travel	26
2.4 Comparación entre aplicaciones	27
Capítulo 3: PLANIFICACIÓN	29
3.1 Metodología de trabajo	29
3.2 Planificación temporal	30
3.3 Análisis	31
3.3.1 Componentes Hardware	31
3.3.2 Componentes Software	32
3.3.3 Personal	32
3.3.4 Método de puntos de función (Método <i>Albretch</i>)	34
3.3.5 Estimación por COCOMO	38
3.3.6 Comparativa entre estimaciones	41
3.3.7 Presupuesto total	41
3.3.8 Seguimiento real	41
Capítulo 4: ANÁLISIS DEL SISTEMA	45
4.1 Descripción de los actores	45
4.2 Requisitos de usuario	45
4.3 Casos de uso	46
4.4 Especificación de casos de uso	48
4.5 Requisitos de información	66
4.5.1 Diagrama Entidad-Relación	66
4.5.2 Diccionario de datos	69
Capítulo 5: DISEÑO	71

5.1 Arquitectura lógica.....	71
5.2 Arquitectura física.....	73
5.3 Diseño del modelo de datos	75
5.4 Diagrama de clases.....	76
5.5 Diagrama de secuencia.....	78
5.6 Diseño de interfaz de usuario.....	80
Capítulo 6: IMPLEMENTACIÓN.....	90
6.1 Guía rápida para desplegar aplicaciones basadas en Vue.js.....	90
6.2 Autenticación y creación de cuentas de usuario.....	93
6.3 Creación de viajes y sus participantes.....	94
6.4 Inserción de gastos y cálculo de los costes	95
6.4 Gestión de actividades	96
Capítulo 7: PRUEBAS	97
7.1 Pruebas de caja blanca	97
7.2 Pruebas de caja negra.....	97
Capítulo 8: DOCUMENTACIÓN DEL PRODUCTO	103
8.1 Manual de instalación	103
8.1.1 Instalación y despliegue de aplicación.....	103
8.1.2 Ejecución de la aplicación <i>TripMerge</i>	104
8.2 Manual de usuario.....	104
8.2.1 Creación de cuenta e inicio de sesión.....	104
8.2.2 Gestión de viajes	106
8.2.3 Gestión de gastos.....	109
8.2.4 Gestión de Equipaje	110
8.2.5 Gestión de agenda de actividades.....	113
8.2.6 Modificar datos del usuario.....	116
Capítulo 9: CONCLUSIONES Y MEJORAS	117
Capítulo 10: REFERENCIAS	119

Índice de figuras

Ilustración 1 - Diagrama Árbol de Características	18
Ilustración 2 - Logo Visual Studio Code	20
Ilustración 3 - Logo GitHub	20
Ilustración 4 - Logo MYSQL	21
Ilustración 5 - Logo MySQL WorkBench	21
Ilustración 6 - Logo NPM.....	22
Ilustración 7 - Logo Microsoft Word.....	22
Ilustración 8 - Logo Draw.io.....	23
Ilustración 9 - Logo Balsamiq.....	23
Ilustración 10 - Logo de Node.js	24
Ilustración 11 - Logo StarUML	24
Ilustración 12: Logo de TravelPerk	25
Ilustración 13: Logo de Expensya	26
Ilustración 14: Logo de Google Trips.....	27
Ilustración 15 - Fases de los Sprints	29
Ilustración 16 - Tabla de la planificación temporal	30
Ilustración 17 - Diagrama de Gantt	31
Ilustración 18 - Seguimiento real.....	42
Ilustración 19 - Diagrama de Gantt real	42
Ilustración 20 - Diagrama Casos de Uso	47
Ilustración 21 - Diagrama Entidad – Relación	67
Ilustración 22 - Diagrama de la Arquitectura Lógica	71
Ilustración 23 - Patrón Modelo Vista Controlador	72
Ilustración 24 - Diagrama de despliegue	74
Ilustración 25 - Diagrama de clases.....	77
Ilustración 26 - Diagrama de secuencia de login.....	78
Ilustración 27 - Diagrama de secuencia creación viaje	79
Ilustración 28 - Interfaz login	80
Ilustración 29 - Interfaz registro	80
Ilustración 30 - Interfaz mis viajes	81
Ilustración 31 - Interfaz datos de un viaje	81
Ilustración 32 - Interfaz creación o modificación viaje.....	82
Ilustración 33 - Interfaz gastos	82
Ilustración 34 - Interfaz datos de un gasto.....	83
Ilustración 35 - Interfaz creación o modificación gasto	84
Ilustración 36 - Interfaz agenda.....	84
Ilustración 37 - Interfaz calendario.....	85
Ilustración 38 - Consulta actividad.....	86
Ilustración 39 - Interfaz creación o modificación de actividad	86
Ilustración 40 - Interfaz equipaje.....	87
Ilustración 41 - Interfaz artículos.....	88
Ilustración 42 - Interfaz añadir artículos.....	89
Ilustración 43 - Interfaz editar usuario	89
Ilustración 44 - Opciones de creación del proyecto	90
Ilustración 45 - Comandos de creación del proyecto	91
Ilustración 46 - Distribución ficheros back-end.....	92

Ilustración 47 - Distribución ficheros front-end.....	93
Ilustración 48 - Ejemplo creación viajes	94
Ilustración 49 - Ejemplo inserción de gasto	95
Ilustración 50 - Ejemplo gestión de actividades.....	96
Ilustración 51 - MU - Creación de cuenta de usuario.....	105
Ilustración 52 - MU - Inicio de sesión.....	105
Ilustración 53 - MU - Pantalla principal vacía	106
Ilustración 54 - MU - Creación de un viaje.....	106
Ilustración 55 - MU - Pantalla principal con datos.....	107
Ilustración 56 - MU - Datos de un viaje	107
Ilustración 57 - MU - Eliminación de un viaje.....	108
Ilustración 58 - MU - Editar viaje	108
Ilustración 59 - MU - Vista de gastos	109
Ilustración 60 - MU - Gastos de un viaje.....	109
Ilustración 61 - MU - Editar gasto.....	110
Ilustración 62 - MU - Vista de equipajes.....	110
Ilustración 63 - MU - Artículos de un equipaje.....	111
Ilustración 64 - MU - Añadir artículos	112
Ilustración 65 - MU – Vista de agenda	113
Ilustración 66 - MU - Calendario de actividades.....	113
Ilustración 67 - MU - Crear actividad	114
Ilustración 68 - MU - Datos de una actividad	114
Ilustración 69 - MU - Crear actividad	115
Ilustración 70 - MU - Editar actividad.....	115
Ilustración 71 - MU - Editar usuario en menú de navegación.....	116
Ilustración 72 - MU - Editar usuario	116
Ilustración 73 - MU - Eliminar usuario	116

Índice de tablas

Tabla 1 - Objetivos del proyecto	15
Tabla 2 - Criterios de Aceptación para OBJ-01	15
Tabla 3- Criterios de Aceptación para OBJ-02	16
Tabla 4- Criterios de Aceptación para OBJ-03	16
Tabla 5- Criterios de Aceptación para OBJ-04	16
Tabla 6- Criterios de Aceptación para OBJ-05	17
Tabla 7 - Comparativa entre distintas aplicaciones	27
Tabla 8 - Coste de componentes Hardware	31
Tabla 9 - Coste de componentes Software	32
Tabla 10 - Coste de personal	33
Tabla 11 - Tabla de grado de complejidad para ALI (Archivos internos) y AIE (Archivos externos)	34
Tabla 12 - Tabla de grado de complejidad para EE (Entradas) y CE (Consultas)	35
Tabla 13 - Tabla de grado de complejidad para SE	35
Tabla 14 - Entradas del usuario	35
Tabla 15 - Salidas del usuario	35
Tabla 16 - Consultas del usuario	35
Tabla 17 - Ficheros internos (ALI)	36
Tabla 18 - Índices de ponderación	36
Tabla 19 - Puntos de función sin ajustar	36
Tabla 20 - Elementos valorables para cálculo factor de ajuste	37
Tabla 21 - Costes de personal (Puntos de función)	38
Tabla 22 - Valor de los factores	39
Tabla 23 - Factor de ajuste COCOMO	39
Tabla 24 - Ponderaciones de modo orgánico	40
Tabla 25 - Costes de personal (COCOMO)	40
Tabla 26 - Requisitos de usuario del proyecto	45
Tabla 27 - Especificación de CU-01	48
Tabla 28 - Especificación de CU-02	48
Tabla 29 - Especificación de CU-03	49
Tabla 30 - Especificación de CU-04	49
Tabla 31 - Especificación de CU-05	50
Tabla 32 - Especificación de CU-06	51
Tabla 33 - Especificación de CU-07	52
Tabla 34 - Especificación de CU-08	53
Tabla 35 - Especificación de CU-09	53
Tabla 36 - Especificación de CU-10	54
Tabla 37 - Especificación de CU-11	55
Tabla 38 - Especificación de CU-12	56
Tabla 39 - Especificación de CU-13	57
Tabla 40 - Especificación de CU-14	58
Tabla 41 - Especificación de CU-15	59
Tabla 42 - Especificación de CU-16	60
Tabla 43 - Especificación de CU-17	61
Tabla 44 - Especificación de CU-18	62
Tabla 45 - Especificación de CU-16	63

Tabla 46 - Especificación de CU-20	64
Tabla 47 - Especificación de CU-21	65
Tabla 48 - Especificación de CU-22	65
Tabla 49 - Especificación de CU-23	66
Tabla 50 - Requisitos de información del proyecto	66
Tabla 51 - Entidad Usuario	69
Tabla 52 - Entidad Actividad	69
Tabla 53 - Entidad Equipaje	69
Tabla 54 - Entidad Artículo	69
Tabla 55 - Entidad Gasto	70
Tabla 56 - PCN-01 Crear cuenta de usuario	98
Tabla 57 - PCN-02 Iniciar sesión	98
Tabla 58 - PCN-03 Añadir un viaje	98
Tabla 59 - PCN-04 Modificar datos de un viaje	99
Tabla 60 - PCN-05 Eliminar un viaje	99
Tabla 61 - PCN-06 Añadir una actividad	99
Tabla 62 - PCN-07 Modificar datos de una actividad	100
Tabla 63 - PCN-08 Eliminar una actividad	100
Tabla 64 - PCN-09 Añadir un gasto	100
Tabla 65 - PCN-10 Modificar un gasto	101
Tabla 66 - PCN-11 Eliminar un gasto	101
Tabla 67 - PCN-12 Añadir artículos en el equipaje	101
Tabla 68 - PCN-13 Eliminar un artículo	102
Tabla 69 - PCN-14 Modificar datos de un usuario	102
Tabla 70 - PCN-15 Eliminar un usuario	102

Capítulo 1: INTRODUCCIÓN

1.1 Organización del documento

En este primer apartado se procede a describir la manera en la que se va a organizar la documentación del proyecto. Para ello, se va a dividir la memoria en nueve partes, con varios índices para facilitar la lectura. A continuación, se procede a explicar brevemente los contenidos de los capítulos:

- **Capítulo 1-Introducción:** Apartado donde se describen los aspectos más generales del proyecto como son introducción, motivación, objetivos y alcance.
- **Capítulo 2-Estado del arte:** Apartado donde se realiza el estudio sobre las distintas aplicaciones, ya creadas anteriormente, las cuales tienen un fin común con este proyecto.
- **Capítulo 3-Planificación y presupuestos:** Apartado donde se analiza la metodología de trabajo aplicada para poder llevar a cabo el proyecto junto con la planificación temporal y finalizada con la estimación del presupuesto total.
- **Capítulo 4-Análisis del sistema:** Apartado donde se describen los requisitos proporcionados por el cliente para la funcionalidad de la aplicación, unido también con la descripción de los distintos actores.
- **Capítulo 5-Diseño e implementación:** Apartado donde se encuentran las arquitecturas físicas y lógicas del sistema, las distintas interfaces y los correspondientes aspectos técnicos.
- **Capítulo 6-Pruebas:** Apartado donde se muestran todas las pruebas pertenecientes al proyecto para comprobar el correcto funcionamiento de este.
- **Capítulo 7-Documentación del producto:** Apartado donde se explican las instrucciones para poder utilizar la aplicación web de una manera óptima. Se detallan todas las posibles funcionalidades de esta.
- **Capítulo 8-Conclusiones y mejoras:** Apartado donde se lleva a cabo un análisis del proyecto ya finalizado para poder llegar a una conclusión y obtener posibles mejoras a realizar en un futuro.
- **Capítulo 9-Referencias:** Apartado donde se recogen todas las referencias utilizadas para realizar este proyecto.

1.2 Introducción

En la actualidad, el turismo se ha integrado de manera significativa en nuestra rutina diaria, convirtiéndose en una parte intrínseca de nuestras vidas, aunque a veces pasemos por alto su presencia. Si reflexionamos sobre el pasado, podremos apreciar cómo la forma de viajar ha experimentado transformaciones significativas gracias a la globalización y el avance de las nuevas tecnologías. En la actualidad, realizar excursiones, tanto a nivel nacional como internacional, se ha vuelto considerablemente más sencillo y accesible. Las opciones para desplazarse son variadas, ya sea en automóvil, autobús o avión, y todas ellas pueden reservarse cómodamente desde el sofá de nuestra casa con un simple clic.

Esta facilidad para viajar se ha visto facilitada por el aumento de la oferta de hoteles y hostales en diversos países, así como por el surgimiento de empresas de transporte de bajo coste. Estos dos elementos han desempeñado un papel crucial en la evolución actual del turismo. Aunque la pandemia mundial del año 2020 generó una drástica disminución en el número de turistas debido a las restricciones y cuarentenas obligatorias, actualmente la situación ha vuelto a la normalidad, evidenciándose una mejora significativa en las estadísticas de los viajes a nivel mundial.

En la actualidad, organizar viajes o excursiones en compañía de amigos, pareja o compañeros de trabajo es una práctica habitual. Es precisamente para satisfacer esta creciente demanda que surge este proyecto, dirigido a aquellas personas que desean gestionar y planificar sus escapadas en grupo, compartiendo así estas experiencias con los demás.

1.3 Motivación

A raíz del deseo de emprender viajes en compañía y gracias a la notable informatización de los aspectos cotidianos en nuestra vida, surge la necesidad imperante de desarrollar una aplicación web que simplifique estas experiencias.

El propósito fundamental de este trabajo de fin de grado es la creación de la mencionada aplicación web, diseñada específicamente para atender las necesidades de aquellos individuos que buscan optimizar sus viajes en grupo o de manera individual. Si bien se espera que los grupos de amigos sean los principales beneficiarios de esta aplicación, también se contempla la posibilidad de que personas que realicen excursiones de manera individual encuentren en ella una herramienta valiosa.

1.4 Objetivos

El objetivo principal de *TripMerge* es unificar las partes de un viaje entre distintas personas en una sola aplicación web. Para poder llevar a cabo dicho objetivo, se reúnen varios secundarios en la siguiente tabla:

ID Objetivo	Nombre Objetivo
OBJ-01	Permitir a los usuarios planificar y organizar itinerarios detallados para sus viajes.
OBJ-02	Posibilitar la creación y gestión de grupos de viaje, donde los participantes puedan colaborar en la planificación.
OBJ-03	Guardar reservas de transporte, alojamiento, etc. Previamente realizadas por los usuarios.
OBJ-04	Gestionar los gastos producidos por los usuarios registrados en el viaje.
OBJ-05	Gestionar el equipaje de los usuarios de cada uno de los viajes para obtener un mejor control de ellos.

Tabla 1 - Objetivos del proyecto

Tras haber listado el conjunto de objetivos de este trabajo, se citarán, a continuación, los criterios de aceptación de cada uno de ellos para garantizar el óptimo cumplimiento de dicho objetivo:

ID Criterio Aceptación	ID Objetivo asociado	OBJ-01
	Descripción Criterio de Aceptación	
CA-01	La aplicación permitirá crear viajes.	
CA-02	La aplicación almacenará los datos de cada uno de los viajes creados.	
CA-03	La aplicación permitirá observar los datos de los viajes a los usuarios.	
CA-04	La aplicación permitirá eliminar viajes.	

Tabla 2 - Criterios de Aceptación para OBJ-01

ID Criterio Aceptación	ID Objetivo asociado	OBJ-02
	Descripción Criterio de Aceptación	
CA-01	La aplicación permitirá compartir los viajes a distintos usuarios.	
CA-02	La aplicación permitirá eliminar a usuarios de un viaje determinado.	
CA-03	La aplicación gestionará las cuentas de los usuarios.	
CA-04	Los usuarios podrán proporcionar una actividad en un día concreto del itinerario.	

Tabla 3- Criterios de Aceptación para OBJ-02

ID Criterio Aceptación	ID Objetivo asociado	OBJ-03
	Descripción Criterio de Aceptación	
CA-01	La aplicación permitirá almacenar datos de las reservas previamente realizadas para el viaje.	
CA-02	La aplicación permitirá eliminar los datos de las reservas.	
CA-03	La aplicación permitirá modificar los datos de las reservas.	

Tabla 4- Criterios de Aceptación para OBJ-03

ID Criterio Aceptación	ID Objetivo asociado	OBJ-04
	Descripción Criterio de Aceptación	
CA-01	La aplicación contará con un apartado para gestionar los gastos realizados en el viaje.	
CA-02	El usuario podrá introducir un gasto en la aplicación.	
CA-03	El usuario podrá modificar un gasto en la aplicación.	
CA-04	El usuario podrá eliminar un gasto en la aplicación.	
CA-05	La aplicación calculará el gasto individual de cada usuario.	

Tabla 5- Criterios de Aceptación para OBJ-04

ID Criterio Aceptación	ID Objetivo	OBJ-05
	asociado	
Descripción Criterio de Aceptación		
CA-01	La aplicación contará con un apartado para almacenar el equipaje con todos sus artículos.	
CA-02	El usuario podrá introducir los artículos necesarios en la aplicación.	
CA-03	El usuario podrá eliminar los artículos previamente introducidos.	
CA-04	El usuario podrá modificar los artículos previamente introducidos.	

Tabla 6- Criterios de Aceptación para OBJ-05

1.5 Alcance

Este proyecto va dedicado especialmente a grupos de amigos o conocidos que se han puesto de acuerdo para realizar un viaje a un destino en concreto.

Esta aplicación les facilitará la tarea de gestión y organización, ya que, al juntarse varias personas para llevar a cabo una actividad, en algunas ocasiones es bastante complicado y laborioso. Para ello, el objetivo principal de este proyecto es unificar todas las acciones necesarias para hacer ese viaje de una manera óptima, en una sola aplicación.

Pero no solo está dirigida a este ámbito, sino que se puede aplicar al sector profesional, ya que existen muchos viajes de negocios dirigidos por las empresas. Gracias a esta aplicación, dichas empresas podrán organizar las actividades que debe desarrollar los trabajadores en cuestión.

Para comenzar, este proyecto estará desarrollado solo como aplicación web, pero un requisito para el futuro será crear una aplicación para Android y iOS para facilitar aún más su usabilidad.

Para visualizar de una mejor manera las características principales de la aplicación, se mostrará a continuación un árbol de características:

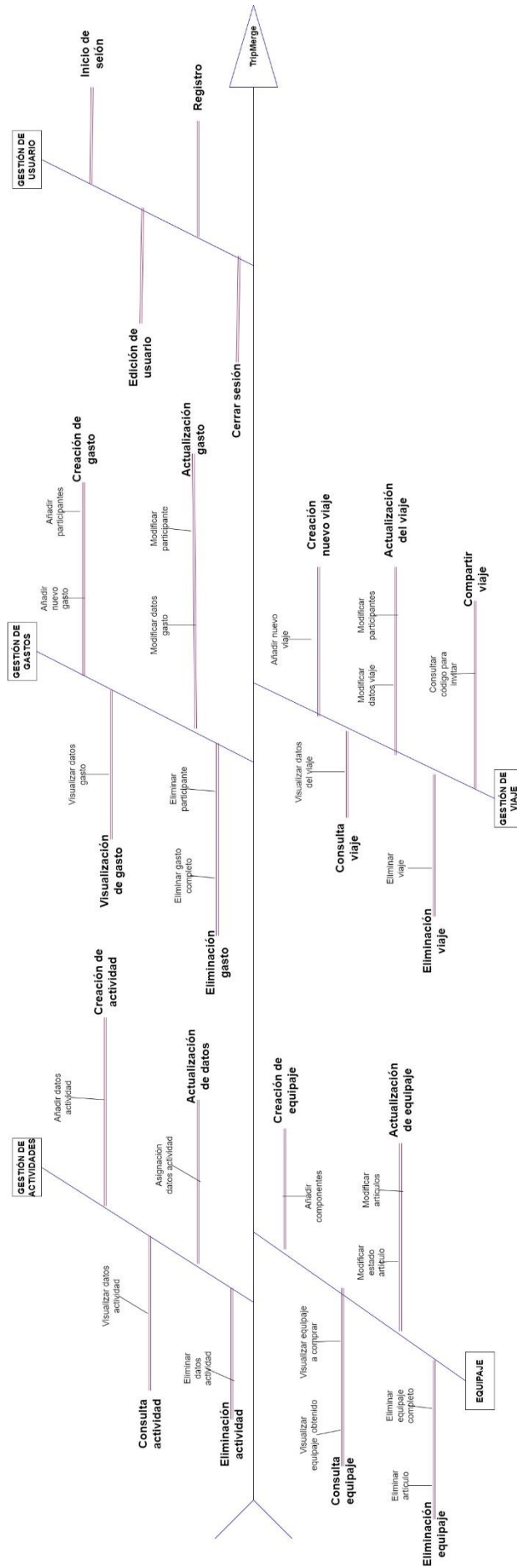


Ilustración 1 - Diagrama Árbol de Características

Como se puede observar en el diagrama anterior, se cuenta con cuatro ámbitos en el que el usuario puede interactuar:

- **Gestión de viaje:** Es la parte más importante del proyecto, la que engloba todas las demás. Permite al usuario crear, modificar o eliminar un viaje para poder trabajar con él más adelante. Al introducir un nuevo viaje en la aplicación, se podrá consultar todos sus datos en cualquier momento a través de la interfaz de usuario. Y, también, el usuario podrá obtener el código del viaje para, posteriormente, compartirlo con otras personas para añadir nuevos participantes.
- **Gestión de gastos:** Tras haber introducido el viaje en la aplicación, el usuario podrá almacenar los gastos realizados por una o varias personas para llevar un control exacto de ellos de una manera sencilla. Se permitirá modificar los datos y los participantes y marcar un gasto como pagado para controlar posibles deudas.
- **Gestión de actividades:** El usuario podrá gestionar las actividades a para organizar de una manera clara y sencilla la excursión. Con esta funcionalidad, lo que se consigue es evitar olvidos que pueden llegar a fastidiar un viaje que lleva un tiempo planeándose.
- **Equipaje:** La aplicación permitirá al usuario almacenar su equipaje para prevenir el descuido de no llevar algún artículo importante que más adelante puede complicar la estancia en el sitio. Este apartado, contendrá una lista con los productos que ya ha obtenido el usuario y una lista de la compra para gestionar qué falta de adquirir antes de viajar.
- **Gestión de usuario:** Por último, el usuario deberá crearse una cuenta en el registro para utilizar las funcionalidades de la aplicación. Una vez registrado, podrá iniciar sesión en cualquier momento utilizando su email y contraseña. Y, dentro de la aplicación, podrá modificar todos sus datos y cerrar la sesión.

1.6 Entorno tecnológico

A continuación, se describirán brevemente las aplicaciones y herramientas utilizadas para poder desarrollar de una manera óptima la aplicación de este proyecto:

- **VISUAL STUDIO CODE**

VISUAL STUDIO CODE (VS CODE) es un editor de código fuente desarrollado por Microsoft. Es un entorno de desarrollo liviano, gratuito y de código abierto que admite una amplia gama de lenguajes de programación. Está disponible para WINDOWS, MACOS y LINUX. Ha ganado popularidad en la comunidad de desarrollo debido a su flexibilidad, rendimiento y la comunidad activa que contribuye con extensiones y mejoras continuas.



Ilustración 2 - Logo Visual Studio Code

- **GITHUB**

GITHUB es una plataforma de desarrollo colaborativo basada en la nube que utiliza el sistema de control de versiones Git. Fue lanzada en 2008 y se ha convertido en una de las plataformas más populares para alojar proyectos de software, colaborar en el desarrollo de código, realizar un seguimiento de cambios en el código fuente y gestionar proyectos de manera eficiente.

En este proyecto, GITHUB se ha utilizado como gestor de versiones para almacenar los cambios y evoluciones del código.



Ilustración 3 - Logo GitHub

- **MySQL**

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. presenta algunas ventajas que lo hacen muy interesante para los desarrolladores. La más evidente es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.



Ilustración 4 - Logo MYSQL

- **MYSQL WORKBENCH**

MYSQL WORKBENCH es una herramienta gráfica de diseño y administración de bases de datos para MYSQL. Es desarrollada por Oracle y proporciona una interfaz visual que facilita la creación, diseño y administración de bases de datos MYSQL.



Ilustración 5 - Logo MySQL WorkBench

- **NPM**

NPM (NODE PACKAGE MANAGER) es el sistema de gestión de paquetes para Node.js, un entorno de ejecución de JavaScript en el lado del servidor. NPM se utiliza para instalar, compartir y gestionar las dependencias de proyectos desarrollados con Node.js. NPM es una herramienta esencial en el ecosistema de Node.js, facilitando la gestión de dependencias y la distribución de paquetes, lo que contribuye a la eficiencia y colaboración en el desarrollo de proyectos Node.js.



Ilustración 6 - Logo NPM

- **MICROSOFT WORD**

MICROSOFT WORD es parte del paquete de software de productividad Microsoft Office. Es una aplicación de procesamiento de textos que permite a los usuarios crear, editar y formatear documentos de texto. Con esta aplicación se ha conseguido realizar toda la documentación del proyecto.



Ilustración 7 - Logo Microsoft Word

- **DRAW.IO**

DRAW.IO es una plataforma en línea que proporciona una herramienta de diagramación y creación de diagramas. Es conocida por su interfaz de usuario intuitiva y la capacidad de crear una variedad de diagramas, desde diagramas de flujo y organigramas hasta diagramas de red y más. Todos los diagramas que conforman el proyecto han sido posibles gracias a draw.io, ya que la mayoría de los programas para crear diagramas online son de pago.



Ilustración 8 - Logo Draw.io

- **BALSAMIQ**

BALSAMIQ es un programa especializado en la creación de interfaces de usuario. Se ha elegido este software debido a su uso muy sencillo y por su diseño, ya que simula los dibujos a mano alzada. Ofrece una variedad de elementos prediseñados: barras de estado, menús, barras de progreso, etc. Como ventaja se puede destacar que es una herramienta multiplataforma y que posee versión gratuita.



balsamiq Wireframes

Ilustración 9 - Logo Balsamiq

- **NODE.JS:**

NODE.JS es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.



Ilustración 10 - Logo de Node.js

- **STARUML**

STARUML es una herramienta para el modelamiento de software basado en los estándares UML (*Unified Modeling Language*) y MDA (*Model Driven Architecture*). En este proyecto se utiliza principalmente para realizar el diagrama de clases ya que facilita mucho su creación debido a su intuitivo y sencillo programa.



Ilustración 11 - Logo StarUML

Capítulo 2: ESTADO DEL ARTE

Para poder conseguir todos los objetivos con un óptimo resultado, es muy importante adquirir los conocimientos y las distintas propuestas de otras aplicaciones que tienen un fin parecido. Para ello, se ha realizado un análisis del mercado para poner en conjunto las ideas principales y compararlas para mejorar todo lo posible.

En el ámbito que ahora nos ocupa, se encuentran en el mercado varias aplicaciones con un objetivo común, que es poder planificar y organizar un viaje. Aunque, la mayoría de las aplicaciones se centran única y exclusivamente en buscar y reservar hoteles o actividades en las ciudades correspondientes. Y es ahí donde se encuentra la mayor diferencia con este proyecto.

A continuación, se citarán y describirán brevemente algunos ejemplos de aplicaciones ya existentes.

2.1 TravelPerk España

TRAVELPERK se encuentra como el ejemplo más parecido al proyecto del mercado actual. Se centra en gestionar viajes de negocios para las empresas, unificando todo en una sola plataforma. Principalmente trata de una aplicación de pago, pero proporciona una demo gratuita para probarlo.

Para ofrecer una gran experiencia al usuario, trabaja con un gran número de empresas para agilizar las reservas y así poder disminuir la dificultad que puede llegar a tener un viaje con tantas personas.

No solo se trata de reservas en hoteles, sino que cuenta también con un gran inventario de empresas de transporte, ya sea trenes, aviones, vehículos de alquiler...

Y para finalizar, cuenta con un gestor de gastos para llevar un control de las facturas durante la experiencia completa.



Ilustración 12: Logo de TravelPerk

2.2 Expensya

EXPENSYA es una aplicación bastante parecida a la anterior ya que ofrece las mismas soluciones.

Se centra también en la gestión de viajes entre los trabajadores de las empresas. Trabaja con varias multinacionales, grandes empresas, PYMES (Pequeña y mediana empresa) y microempresas. Cuenta también con un gestor de gastos, optimiza el ROI (Return On Investment) de los gastos y obtiene una centralización de datos en tiempo real.

Su gran diferencia con TRAVELPERK, es que ofrece una conversión automática de divisas para los viajes a un país extranjero. Es una gran manera de simplificar al máximo la excursión de una empresa.



Ilustración 13: Logo de Expensya

2.3 Google Travel

Por último, y no menos importante, se encuentra la aplicación GOOGLE TRAVEL, creada por la gran compañía GOOGLE. Está claro que, hoy en día, es muy complicado superar a las grandes empresas en cuanto a novedades. Con lo cual, las nuevas aplicaciones se acaban centrando en las ya creadas. En este caso, GOOGLE TRAVEL es de las mejores en cuanto a gestión de viajes en el mercado actual.

Dicha aplicación nos ofrece una interfaz minimalista y cuidada para simplificar al máximo la experiencia de usuario. Como procesos a seguir cuenta con:

- Explorar: Ofrece una información muy variada del destino escogido por el usuario.
- Qué hacer: Apartado para almacenar las distintas actividades a realizar en dicho viaje

- Vuelos: Gestiona el viaje de las distintas formas de transporte posibles de una ciudad a otra, encontrando las ofertas correspondientes.
- Hoteles: Busca los mejores hoteles en la ciudad que introduzca el usuario.
- Alquileres vacacionales: Busca hostales o apartamentos en alquiler.



Ilustración 14: Logo de Google Trips

2.4 Comparación entre aplicaciones

	TRAVELPERK	EXPENSYA	GOOGLE TRAVEL	TRIPMERGE
Cuentas de usuario	✓	✓	✗	✓
Administrar viajes	✓	✓	✓	✓
Agenda de actividades	✗	✗	✗	✓
Gestor de equipaje	✗	✗	✗	✓
Gestor de gastos	✓	✓	✗	✓
Colaboración entre usuarios	✗	✗	✓	✓
Gestor de reservas	✓	✓	✓	✓
Conversión de divisas	✗	✓	✗	✗

Tabla 7 - Comparativa entre distintas aplicaciones

Capítulo 3: PLANIFICACIÓN

3.1 Metodología de trabajo

Antes de desarrollar el proyecto, es imprescindible analizar las distintas metodologías de desarrollo y escoger la más conveniente. En este caso, se ha decidido implementar la metodología ágil (Agile), ya que es una de las cuales se ha hecho más énfasis en la carrera universitaria y en la que tengo más experiencia.

La metodología ágil es un conjunto de técnicas aplicadas en ciclos de trabajo cortos, con el objetivo de que el proceso de entrega de un proyecto sea más eficiente. Así, con cada etapa completada, ya se pueden entregar avances y se deja de lado la necesidad de esperar hasta el término del proyecto.

La metodología ágil Scrum se basa en ciclos de trabajo o *sprints* (que pueden durar desde semanas hasta meses) en los que se entrega alguna parte del proyecto. En cada nuevo *sprint* se genera una versión del producto que supera a la anterior. El pilar fundamental de la metodología ágil Scrum es que todas las decisiones se toman en función de la información existente y de la propia experiencia de los integrantes del equipo.

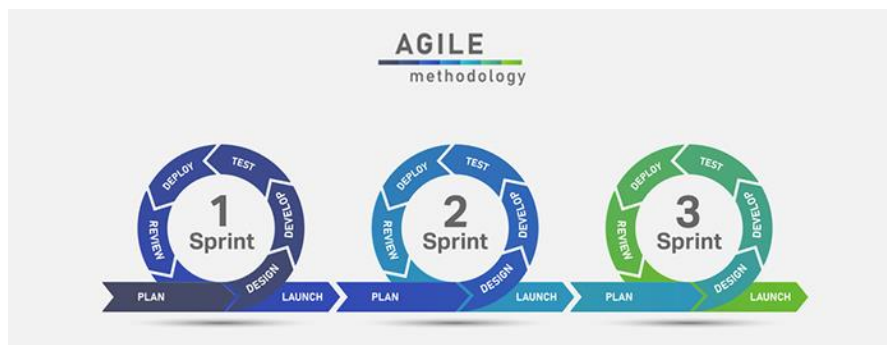


Ilustración 15 - Fases de los Sprints

Como puede verse en la ilustración anterior, cada *sprint* está compuesto de cinco etapas que se describen brevemente a continuación:

1. **Diseño - Planificación:** El *sprint* comienza con una reunión de planificación, donde los miembros del equipo se reúnen para establecer los componentes para la próxima ronda de trabajo.
2. **Desarrollo:** Desarrollo del producto de acuerdo con las pautas aprobadas.
3. **Pruebas:** Pruebas exhaustivas completas y documentación de los resultados antes de la implementación.
4. **Despliegue:** Presentación del producto o software en funcionamiento a las partes interesadas y los clientes.
5. **Revisión:** Solicitud de comentarios al cliente y a las partes interesadas y recopilación de información para incorporarla en el próximo *sprint*.

Este método, además, incluye la realización de reuniones con el cliente. En este caso, se realizarán las reuniones con el tutor del proyecto, ya que no existe un cliente externo. En dichas reuniones, se presentará la memoria y, si existe, parte del desarrollo de la aplicación. Cada reunión corresponderá a un *sprint* diferente, en el que se corregirán los fallos descritos en el *sprint* anterior y se completará una parte nueva para su posterior corrección. Tras varios *sprints*, se llegará al último para garantizar la finalización del proyecto.

3.2 Planificación temporal

Nombre de tarea	Duración	Comienzo	Fin
Desarrollo de frontend sin datos	28 días	lun 15/01/24	mié 21/02/24
Planificación y diseño	12 días	lun 15/01/24	mar 30/01/24
Desarrollo	8 días	mié 31/01/24	vie 09/02/24
Pruebas	2 días	lun 12/02/24	mar 13/02/24
Despliegue	2 días	mié 14/02/24	jue 15/02/24
Revisión	4 días	vie 16/02/24	mié 21/02/24
Desarrollo de backend	26 días	jue 22/02/24	jue 28/03/24
Planificación y diseño	10 días	jue 22/02/24	mié 06/03/24
Desarrollo	8 días	jue 07/03/24	lun 18/03/24
Pruebas	2 días	mar 19/03/24	mié 20/03/24
Despliegue	2 días	jue 21/03/24	vie 22/03/24
Revisión	4 días	lun 25/03/24	jue 28/03/24
Desarrollo de frontend con datos	27 días	vie 29/03/24	lun 06/05/24
Planificación y diseño	11 días	vie 29/03/24	vie 12/04/24
Desarrollo	10 días	lun 15/04/24	vie 26/04/24
Pruebas	2 días	lun 29/04/24	mar 30/04/24
Despliegue	1 día	mié 01/05/24	mié 01/05/24
Revisión	3 días	jue 02/05/24	lun 06/05/24

Ilustración 16 - Tabla de la planificación temporal

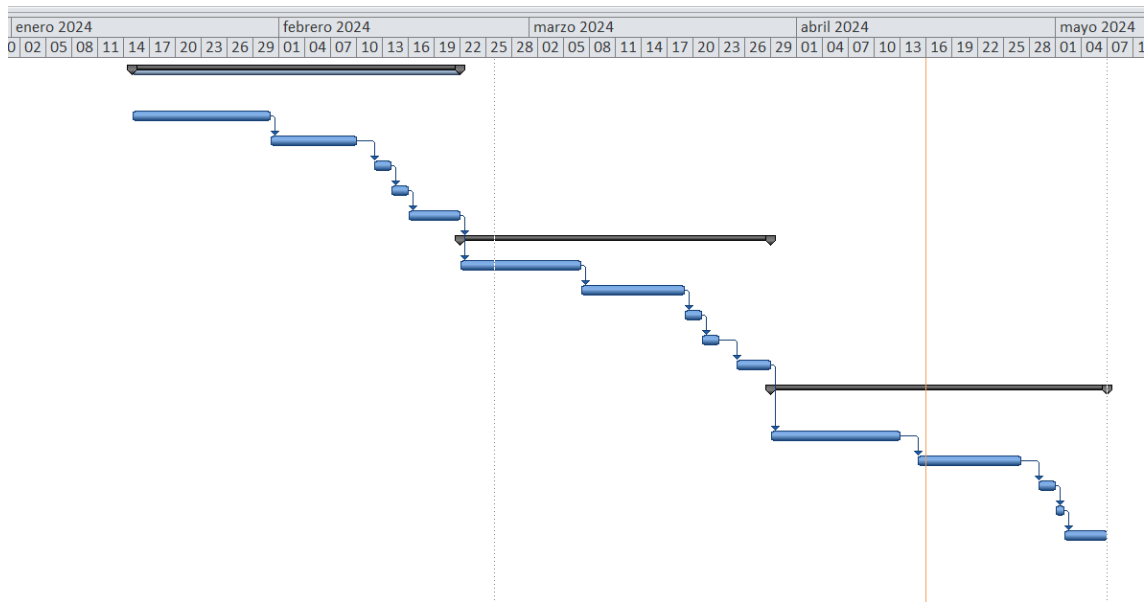


Ilustración 17 - Diagrama de Gantt

3.3 Análisis

3.3.1 Componentes Hardware

Para realizar este proyecto, es fundamental obtener unos componentes hardware para llevar a cabo, tanto la documentación como la parte de código. Todos los componentes fueron adquiridos antes del comienzo de este trabajo, pero sólo se analizarán los cuatro meses que se calcula como tiempo total. Al utilizar solo los cuatro meses, el porcentaje de uso será reducido y, por tanto, el coste real también. A continuación, se mostrará una tabla con la información de cada uno de los componentes hardware utilizados por el dueño del proyecto:

Componente	Precio	Vida útil (años)	% uso	Coste real
Ordenador	700 €	5	6,66 %	46,62 €
Monitor	150 €	4	8,33 %	12,50 €
Ratón	30 €	3	11,11 %	3,33 €
Teclado	40 €	4	8,33 %	3,33 €
Internet	55 €	1	400,00 %	220,00 €
Coste total				285,78 €

Tabla 8 - Coste de componentes Hardware

Como se puede observar en la tabla anterior, el coste total de todos los componentes Hardware utilizados para este proyecto es 285,78€. Lo que hace que aumente tanto el coste es el precio del Internet del hogar ya que se paga mes a mes y su porcentaje de uso llega hasta el 400%.

3.3.2 Componentes Software

Una vez adquiridos todos los componentes Hardware, toca el turno del Software. Para ello se mostrará a continuación una tabla con toda su información. Para calcular el porcentaje de uso se utilizará el mismo criterio que ya se ha comentado en el apartado anterior, asumir cuatro meses del total de vida útil del recurso para estimar el coste asociado del mismo.

Componente	Precio	Vida útil (años)	% uso	Coste real
WINDOWS 11	170 €	5	6,66 %	11,32 €
MICROSOFT OFFICE	69 €	1	33,33 %	22,99 €
VISUAL STUDIO CODE	0 €			
GITHUB	0 €			
MYSQL WORKBENCH	0 €			
DRAW.IO	0 €			
BALSAMIQ	0 €			
Coste total				34,31 €

Tabla 9 - Coste de componentes Software

Comparando los costes de los componentes Hardware y Software vemos una gran diferencia ya que para la parte de Software la mayoría de los programas que se utilizan son de uso gratuito.

3.3.3 Personal

Finalmente, junto con los componentes anteriores, es necesario tener personal que pueda ejercer su función en el proyecto. En general, lo primordial sería contratar a varias personas dependiendo su puesto, pero en este caso, al ser un trabajo de fin de carrera, se ha realizado solamente por una persona. Esa persona hace el puesto de varios roles distintos:

- **Jefe de proyecto:** Encargado de desarrollar un plan detallado del proyecto, definiendo sus objetivos, alcance, recursos, etc. Para ello necesita también controlar a los empleados y organizarles en distintas tareas.
- **Analista:** Encargado de reunir todos los requisitos del negocio y traducirlos en distintas especificaciones.
- **Desarrollador:** Encargado de escribir y mantener el código de la aplicación web.

Una vez definidos los distintos roles, a continuación, se realizará un análisis de todos los costes con un sueldo hipotético y una duración de cuatro meses.

Para comenzar, se tiene en cuenta que para este proyecto se trabajará a media jornada, es decir, veinte horas semanales. Teniendo cuatro meses y veinte horas semanales, hacen un resultado de 320 horas de duración total para la aplicación.

En la Universidad, el Trabajo de Fin de Grado está calculado como 12 ECTS (Créditos) y cada ECTS equivale a un rango de 25-30 horas. Entonces, se obtiene un rango total de 300 y 360 horas de duración. Este proyecto está estimado en 320 horas, con lo cual entra en el rango de la Universidad.

Una vez citado todos los roles y el total de horas del proyecto, se procede a repartir las horas de trabajo dependiendo del rol que se ejerza.

- **Jefe de proyecto:** 35 % de 320 horas = 112 horas
- **Analista:** 25 % de 320 horas = 80 horas
- **Desarrollador:** 40 % de 320 horas = 128 horas

A continuación, tras una búsqueda en fuentes de Internet (citada en el capítulo 9 de Referencias), se especifican los sueldos brutos por hora.

- **Jefe de proyecto:** 19,23 € / hora
- **Analista:** 14,36 € / hora
- **Desarrollador:** 13,62 € / hora

Pero para este tipo de proyecto, lo que interesa sería el salario neto, por lo que habría que reducirlos aproximadamente un 30 %, debido a los siguientes gastos laborales:

1. 23,60 % para la seguridad social
2. 5,50 % retención para contrato indefinido
3. 0,20 % de FOGASA
4. 0,70 % para formación profesional) para obviar los impuestos.

Por lo tanto, se procede a aplicar dicho porcentaje a los sueldos brutos:

- **Jefe de proyecto:** $19,23 \text{ €} * 0,7 = 13,46 \text{ €} / \text{hora}$
- **Analista:** $14,36 \text{ €} * 0,7 = 10,05 \text{ €} / \text{hora}$
- **Desarrollador:** $13,62 \text{ €} * 0,7 = 9,53 \text{ €} / \text{hora}$

Finalmente, con el total de horas y los sueldos de los trabajadores, se calcula el coste real de cada uno de ellos:

Rol	Tiempo (horas)	Sueldo/Hora (Neto)	Coste real
Jefe de proyecto	112 h	13,46 €/h	1.507,52 €
Analista	80 h	10,05 €/h	804,00 €
Desarrollador	128 h	9,53 €/h	1.219,84 €
Coste total			3.531,36 €

Tabla 10 - Coste de personal

El coste total de los roles de los trabajadores asciende a 3.531,36 €.

3.3.4 Método de puntos de función (Método *Albretch*)

En este apartado, se procede a analizar la funcionalidad de los sistemas software a través del Método *Albretch*. Consiste en calcular los puntos de función del proyecto con tres etapas:

1. Identificación de los componentes necesarios para el cálculo.
2. Cálculo de los Puntos Función no ajustados.
3. Ajuste de los Puntos Función.

La primera etapa se basa en la identificación de los elementos para tener en cuenta para el cálculo de los puntos de función. Para ello, es necesario obtener los siguientes componentes:

- **Entradas externas (EE):** Se originan de un usuario o se transmiten de otra aplicación.
- **Salidas externas (SE):** Datos que ofrecen información al usuario.
- **Consultas externas (CE):** Entrada en línea que da como resultado la generación de alguna respuesta inmediata.
- **Archivos lógicos internos (ALI):** Agrupamiento de datos que reside en la aplicación y se mantiene mediante las entradas externas.
- **Archivos de interfaz externos (AIE):** Agrupamiento de datos que reside fuera de la aplicación y contiene información que puede usar la aplicación.

Para comenzar con el método hay que clasificar los elementos de cada clase según su grado de complejidad. Para ello, es necesario basarse en el número de Tipos de Datos Elementales (TED) y el número de Tipos de Registros (TR).

- **TED:** Se define como un campo único, no recurrente y reconocible para el usuario en un ALI o AIE.
- **TR:** Se define como un subgrupo de elementos de datos reconocibles para el usuario dentro de un ALI o AIE.

Existen varias tablas para saber el grado de complejidad de cada elemento, dependiendo del número de tipos elementales (TED) y el número de registros (TR):

ALI o AIE	1 a 19 TED	20 a 50 TED	>51 TED
1 TR	Baja	Baja	Media
2 a 5 TR	Baja	Media	Alta
6 o más TR	Media	Alta	Alta

Tabla 11 - Tabla de grado de complejidad para ALI (Archivos internos) y AIE (Archivos externos)

EE y CE	1 a 4 TED	5 a 15 TED	> 15 TED
0-1 TR accedidos	Baja	Baja	Media
2 TR accedidos	Baja	Media	Alta
> 2 TR accedidos	Media	Alta	Alta

Tabla 12 - Tabla de grado de complejidad para EE (Entradas) y CE (Consultas)

SE	1 a 5 TED	6 a 19 TED	> 19 TED
0-1 FR accedidos	Baja	Baja	Media
2-3 FR accedidos	Baja	Media	Alta
> 3 FR accedidos	Media	Alta	Alta

Tabla 13 - Tabla de grado de complejidad para SE

Una vez explicado el primer paso, se procede a analizar qué elementos van a ser utilizados en el proyecto:

Entradas del usuario:

Entradas	Complejidad
Registro	Baja
Inicio de sesión	Baja
Creación de un viaje	Baja
Ingreso de un artículo del equipaje	Baja
Ingreso de un gasto	Baja
Ingreso de una actividad	Media

Tabla 14 - Entradas del usuario

Salidas del usuario:

Salidas	Complejidad
Listado de viajes	Baja
Listado de artículos del equipaje	Baja
Listado de gastos	Baja
Listado de actividades	Media
Datos de un viaje	Baja
Datos de un gasto	Baja
Datos de una actividad	Media

Tabla 15 - Salidas del usuario

Consultas del usuario:

Consultas	Complejidad
Consulta de usuario	Baja
Consulta de viaje	Baja
Consulta de equipaje	Baja
Consulta de gasto	Baja
Consulta de actividad	Media

Tabla 16 - Consultas del usuario

Ficheros internos:

Ficheros internos (ALI)	Complejidad
Archivo de viajes	Baja
Archivo de equipajes	Baja
Archivo de gastos	Baja
Archivo de actividades	Baja
Archivo de usuarios	Baja

Tabla 17 - Ficheros internos (ALI)

Después de haber obtenido los puntos de función sin ajustar (PFNA) de cada componente, se calcula el total valorando el tipo de complejidad que tiene cada uno. En la siguiente tabla se representa de una manera más gráfica los valores que hay que aplicar dependiendo de su complejidad:

	Complejidad baja	Complejidad media	Complejidad alta
Entradas	3	4	6
Salidas	4	5	7
Consultas	3	4	6
Ficheros internos	7	10	15
Ficheros externos	5	7	10

Tabla 18 - Índices de ponderación

A continuación, teniendo en cuenta el número de entradas, salidas, consultas, etc. Y su complejidad, se procede a multiplicarlo por los índices de ponderación de la tabla anterior:

	Complejidad baja	Complejidad media	Complejidad alta	Total
Entradas	$5 \times 3 = 15$	$1 \times 4 = 4$	0	19
Salidas	$5 \times 4 = 20$	$2 \times 5 = 10$	0	30
Consultas	$4 \times 3 = 12$	$1 \times 4 = 4$	0	16
Ficheros internos	$5 \times 7 = 35$	0	0	35
Ficheros externos	0	0	0	0
PFNA totales				100

Tabla 19 - Puntos de función sin ajustar

Una vez calculados los PFNA totales del proyecto, se llega al siguiente paso, analizar el Factor de Ajuste (FA). Consiste en responder del 0 (no importante) al 5 (muy esencial) a los siguientes catorce conceptos:

Elementos valorables para cálculo factor de ajuste	Complejidad
1. Comunicación de datos	4
2. Funciones distribuidas	4
3. Prestaciones	1
4. Gran uso de la configuración	3
5. Velocidad de las transacciones	3
6. Entrada online de datos	3
7. Diseño para la eficiencia del usuario final	3
8. Actualización de datos online	1
9. Complejidad de procesos lógicos internos	1
10. Reusabilidad del código	4
11. Facilidad de instalación	3
12. Facilidad de operación	4
13. Localizaciones múltiples	2
14. Facilidad de cambios	3
Suma valoración complejidad	39

Tabla 20 - Elementos valorables para cálculo factor de ajuste

El último paso para calcular es factor de ajuste sería aplicar la fórmula siguiente, adquirida de los apuntes de Gestión de Proyectos Basados en las TI:

$$FA = 0,65 + 0,01 * Valoración complejidad$$

Siendo la valoración de complejidad previamente calculada, 39:

$$FA = 0,65 + 0,01 * 39 = 1,04 FA$$

El Factor de Ajuste final sería 1,04 FA. Posteriormente, faltaría calcular los puntos de función ajustados (PFA) y para ello se necesitan los PFNA (Puntos de función no ajustados) y el FA (Factor de ajuste), empleados en la siguiente fórmula:

$$PFA = PFNA * FA$$

$$PFA = 100 * 1,04 = 104 PFA$$

Y, para finalizar, se procede a calcular el tiempo total que se estima que va a durar la realización del proyecto. Suponiendo que cada punto de función equivale a cuatro horas de trabajo:

$$\text{Duración del proyecto} = 104 PFA * 4 = 416 \text{ horas de duración}$$

Tras saber el tiempo total estimado, se vuelve a calcular los costes del personal cambiando las horas trabajadas. Para ello, se muestra de nuevo la tabla de costes, utilizando los porcentajes de trabajo según el tiempo estimado que ya se ha utilizado en el apartado 3.3.3.

- Jefe de proyecto: 35 % de 416 = 145,6 horas
- Analista: 25 % de 416 = 104 horas
- Desarrollador: 40 % de 416 = 166,4 horas

Rol	Tiempo (horas)	Sueldo/Hora	Coste real
Jefe de proyecto	145,6 h	13,46 €/h	1.959,78 €
Analista	104 h	10,05 €/h	1.045,20 €
Desarrollador	166,4 h	9,53 €/h	1.585,79 €
Coste total			4.590,77 €

Tabla 21 - Costes de personal (Puntos de función)

Como conclusión de esta estimación, al obtener un número mayor de horas de duración del proyecto, el presupuesto gastado en el coste personal asciende considerablemente hasta un total de 4.590,77 €.

3.3.5 Estimación por COCOMO

En este apartado, se tiene como objetivo calcular la estimación, pero utilizando COCOMO (Modelo Constructivo de Costos). Hay que tener en cuenta que COCOMO incluye tres modos de desarrollo software:

1. **Modo orgánico:** ambiente familiar y estable, el producto es pequeño (menos de 50.000 líneas de código) y requiere pocas innovaciones tecnológicas. Todo el equipo tiene gran experiencia en sistemas relacionados y todos pueden contribuir en las etapas iniciales sin generar confusión.
2. **Modo semi-acoplado:** tamaño de menos de 300.000 líneas de código y complejidad media, con niveles variados de exigencia, algunas partes rigurosas y otras flexibles.
3. **Modo empotrado:** proyectos de gran envergadura en un ambiente complejo con altas restricciones de hardware, software y procedimientos operacionales.

Una vez explicados los distintos modos de desarrollo software, toca el paso de elegir el idóneo para este proyecto. En principio, se descarte el modo empotrado ya que no coincide con las características propias del proyecto. Entre el modo orgánico y el semi-acoplado, se elige el modelo orgánico puesto que es un proyecto pequeño y requiere pocas innovaciones tecnológicas.

Tras haber analizado y elegido un modo de desarrollo, es necesario estimar cuántas líneas de código van a ser necesarias para que la aplicación web funcione de la mejor manera. Como se van a utilizar los lenguajes JavaScript y HTML para el desarrollo, se utiliza una media de 55 líneas de código para cada punto de función no ajustado (Véase apartado 3.3.4). A continuación, se calcula el total de líneas de código:

$$LDC = PFNA * estimación\ de\ líneas$$

$$LDC = 100 * 55 = 5500 \quad LDC = 5,500 \quad KLDC$$

El siguiente paso es aplicar un factor de esfuerzo basado en la siguiente tabla:

Factores	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Fiabilidad requerida	0.75	0.88	1.00	1.15	1.40	
Tamaño de la base de datos		0.94	1.00	1.08	1.16	
Complejidad del software	0.70	0.85	1.00	1.15	1.30	1.65
Restricciones de tiempo de ejecución			1.00	1.11	1.30	1.66
Restricciones de memoria			1.00	1.06	1.21	1.56
Volatilidad del hardware		0.87	1.00	1.15	1.30	
Restricciones de tiempo de respuesta		0.87	1.00	1.07		
Calidad de los analistas	1.46	1.19	1.00	0.86	0.71	
Experiencia con el tipo de aplicación	1.29	1.13	1.00	0.91	0.82	
Experiencia con el hardware	1.21	1.10	1.00	0.90		
Experiencia con el lenguaje de programación	1.14	1.07	1.00	0.95		
Calidad de los programadores	1.42	1.17	1.00	0.86	0.70	
Técnicas modernas de programación	1.24	1.10	1.00	0.91	0.82	
Empleo de herramientas	1.24	1.10	1.00	0.91	0.83	
Restricciones a la duración del proyecto	1.23	1.08	1.00	1.04	1.10	

Tabla 22 - Valor de los factores

I-ésimo Factor ajuste (FA)	Factor	Valor FA
Fiabilidad requerida	Alto	1.15
Tamaño de la base de datos	Medio	1.00
Complejidad del software	Medio	1.00
Restricciones de tiempo de ejecución	Medio	1.00
Restricciones de memoria	Medio	1.00
Volatilidad del hardware	Medio	1.00
Restricciones de tiempo de respuesta	Alto	1.07
Calidad de los analistas	Medio	1.00
Experiencia con el tipo de aplicación	Alto	0.91
Experiencia con el hardware	Alto	0.90
Experiencia con el lenguaje de programación	Alto	0.95
Calidad de los programadores	Medio	1.00
Técnicas modernas de programación	Medio	1.00
Empleo de herramientas	Medio	1.00
Restricciones a la duración del proyecto	Medio	1.00

Tabla 23 - Factor de ajuste COCOMO

Después, con los valores de los factores de ajuste y con el modo elegido, se calcula el factor de ajuste que se aplicará más adelante:

$$\text{Factor de ajuste} = \prod \text{valorFA}(i)$$

$$FA = 1,15 * 1 * 1 * 1 * 1 * 1 * 1,07 * 1 * 0,91 * 0,9 * 0,95 * 1 * 1 * 1 * 1 * 1 = 0,958$$

Una vez calculado el factor de ajuste, hay que calcular el esfuerzo total utilizando los siguientes valores de la tabla:

Modelo	a	b	c	d
Orgánico	3,20	1,05	2,50	0,38

Tabla 24 - Ponderaciones de modo orgánico

$$Esfuerzo (E) = a * (KLDC)^b * FA \text{ [personas * mes]}$$

$$E = 3,20 * (5,5)^{1,05} * 0,958 = 3,20 * 5,99 * 0,958 \text{ [personas * mes]}$$

$$E = 18,36 \text{ personas * mes}$$

Luego, se calcula el tiempo de desarrollo:

$$Tiempo \text{ de desarrollo } (Tdev) = c * E^d$$

$$Tdev = 2,50 * 18,36^{0,38} = 2,50 * 3,02 \text{ meses}$$

$$Tdev = 7,55 \text{ meses}$$

Para llegar a saber el número de personas que son necesarias para realizar el proyecto en 7,55 meses hay que obtener el esfuerzo nominal con la siguiente fórmula:

$$N = E / Tdev \text{ [personas]}$$

$$N = 18,36 / 7,55 = 2,43 \text{ personas}$$

Por lo que, serían necesarias 2,43 personas para llevar a cabo el proyecto en el tiempo estimado de 7,55 meses. En este caso, el proyecto es realizado por una única persona por lo que serían necesarios 18 meses para completar el proyecto, lo que equivale (trabajando media jornada) a 1440 horas de trabajo.

Para finalizar, se calcula el presupuesto total del coste del personal de la misma manera que en el apartado anterior, utilizando los mismos porcentajes:

Rol	Tiempo (horas)	Sueldo/Hora	Coste real
Jefe de proyecto	504 h	13,46 €	6.783,84 €
Analista	360 h	10,05 €	3.618,00 €
Desarrollador	576 h	9,53 €	5.489,28 €
Coste total			15.891,12 €

Tabla 25 - Costes de personal (COCOMO)

Como conclusión de esta estimación, el presupuesto gastado en el coste personal asciende considerablemente a 15.891,12 €.

3.3.6 Comparativa entre estimaciones

Tras haber analizado la estimación de puntos de función y la de COCOMO, se llega a la conclusión que existe un presupuesto total demasiado excesivo en COCOMO. Gracias a ello, se elige la estimación por puntos de función para este proyecto ya que consigue un presupuesto final más realista.

El coste total de puntos de función trata de 4.590,77 €. y el de COCOMO asciende a 15.891,12 €. Por ello, se observa una diferencia de más de 10.000€.

3.3.7 Presupuesto total

Finalmente, para finalizar este capítulo es necesario conocer el presupuesto total para la realización de este proyecto, utilizando la estimación por puntos de función.

Al calcular dicho presupuesto, hay que tener en cuenta los siguientes aspectos:

- **Coste componentes hardware:** 285,78€
- **Coste componentes software:** 34,31€
- **Coste personal:** 5.523,58€

La suma de los anteriores componentes hace un total de 5.843,67 €.

3.3.8 Seguimiento real

Tras haber obtenido los hipotéticos presupuestos analizando el Método *Albretch* y la estimación por COCOMO, es interesante realizar un seguimiento real de este proyecto para observar las diferencias en el caso de que las hubiera. A continuación, se mostrará una imagen con la duración de cada tarea en el seguimiento real del proyecto:

Nombre de tarea	Duración	Comienzo	Fin
Desarrollo de frontend sin datos	26 días	lun 15/01/24	lun 19/02/24
Planificación y diseño	6 días	lun 15/01/24	lun 22/01/24
Desarrollo	15 días	mar 23/01/24	lun 12/02/24
Pruebas	2 días	mar 13/02/24	mié 14/02/24
Despliegue	1 día	jue 15/02/24	jue 15/02/24
Revisión	2 días	vie 16/02/24	lun 19/02/24
Desarrollo de backend	24 días	mar 20/02/24	vie 22/03/24
Planificación y diseño	7 días	mar 20/02/24	mié 28/02/24
Desarrollo	11 días	jue 29/02/24	jue 14/03/24
Pruebas	3 días	vie 15/03/24	mar 19/03/24
Despliegue	1 día	mié 20/03/24	mié 20/03/24
Revisión	2 días	jue 21/03/24	vie 22/03/24
Desarrollo de frontend con datos	25 días	lun 25/03/24	vie 26/04/24
Planificación y diseño	6 días	lun 25/03/24	lun 01/04/24
Desarrollo	12 días	mar 02/04/24	mié 17/04/24
Pruebas	3 días	jue 18/04/24	lun 22/04/24
Despliegue	1 día	mar 23/04/24	mar 23/04/24
Revisión	3 días	mié 24/04/24	vie 26/04/24

Ilustración 18 - Seguimiento real

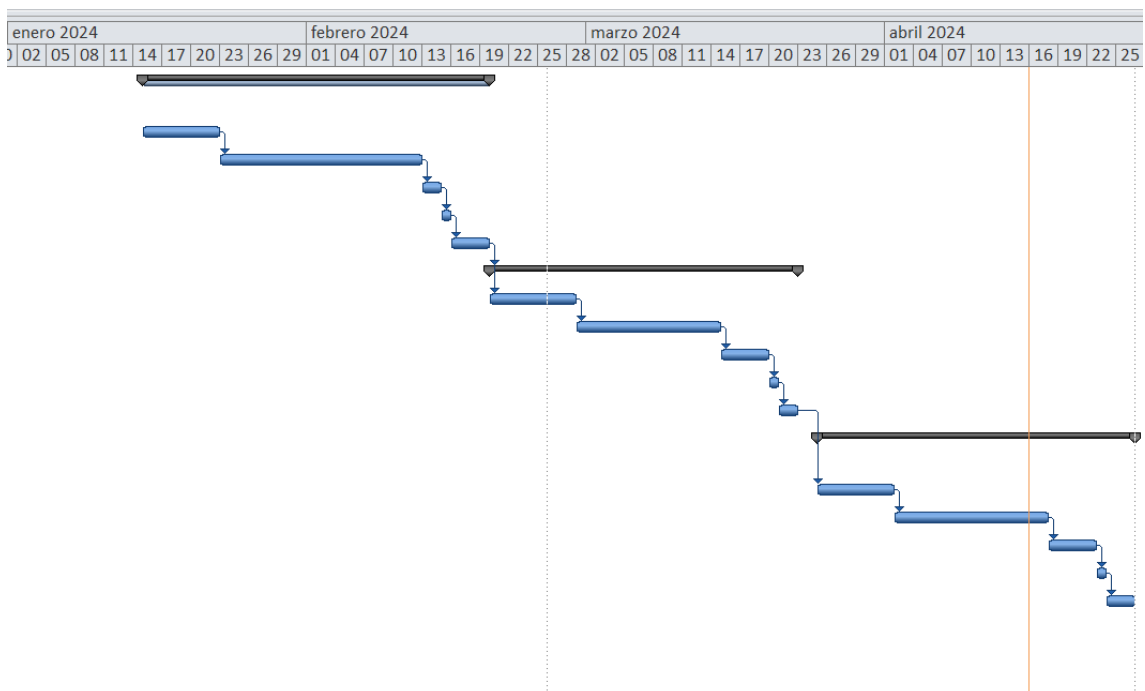


Ilustración 19 - Diagrama de Gantt real

Como se puede observar, el número totales de días han sido reducidos debido a que se hizo una planificación un poco más elevada respecto a su duración real.

Para repartir las tareas entre los roles de los trabajadores se ha considerado la siguiente asignación de roles según qué tarea realizar:

- Planificación y diseño: Jefe de proyecto y analista.
- Desarrollo: Desarrollador.
- Pruebas: Desarrollador.
- Despliegue: Jefe de proyecto.
- Revisión: Analista.

La primera estimación de los días dio como resultado un total de 81 días, en los cuales se trabajaban cuatro horas al día para poder ajustar la duración estimada de un Trabajo de Fin de Grado. Así pues, se consideraba trabajar en un régimen de media jornada y, por lo tanto, se tardarían 320 horas finalmente.

Pero finalmente, observando el seguimiento real, el proyecto ha sido terminado en 75 días, trabajando cuatro horas al día, que hacen un total de 300 horas. Estas horas totales entrarían aproximadamente dentro del rango aportado por los créditos de la Universidad para este tipo de trabajos, ya que es de 300-360 horas estimadas. El reparto de las tareas a los roles han sido las mismas que en la estimación del principio.

- Jefe de proyecto:
 - 13 días en planificación y diseño
 - 3 días en despliegue
- Analista:
 - 6 días en planificación y diseño
 - 7 días en revisión
- Desarrollador:
 - 38 días en desarrollo
 - 8 días en pruebas

Finalmente, gracias al seguimiento real nos damos cuenta de que reducimos la duración final y, por lo tanto, el coste total del proyecto.

Rol	Tiempo (horas)	Sueldo/Hora (Neto)	Coste real
Jefe de proyecto	64 h	13,46 €/h	861,44 €
Analista	52 h	10,05 €/h	522,60 €
Desarrollador	184 h	9,53 €/h	1.753,52 €
Coste total			3.137,56 €

La diferencia respecto a la planificación temporal se justifica mediante la falta de experiencia en el sector de la planificación y diseño, ya que se consideraba que iban a llevar bastantes más días de lo finalmente realizado. Esa diferencia de días, se suman a la falta de horas en el ámbito de la implementación, realizada por el desarrollador.

Como conclusión, el jefe de proyecto y el analista han terminado el trabajo en menos horas de las estimadas, mientras que el desarrollador ha trabajado más. Debido a la diferencia de sus sueldos, el coste total final del proyecto se ha reducido de 3.531,36 € a 3.137,56€

Capítulo 4: ANÁLISIS DEL SISTEMA

4.1 Descripción de los actores

En el ámbito de los casos de uso, se define a un actor cómo alguien o algo fuera del proyecto que interactúa de alguna forma con el sistema.

En este caso, sólo se dispondrá de un actor, el cual será el propio usuario que está utilizando la aplicación. Dicho usuario es el encargado de crearse una cuenta, crear un viaje, crear actividad, etc.

4.2 Requisitos de usuario

Los requisitos de usuario son las expectativas que una organización tiene con respecto al sistema para cumplir con sus necesidades, ya sea de negocio o de cumplimiento. Es el “qué y para qué”.

A continuación, se mostrará un listado del total de requisitos de usuario necesarios para terminar el proyecto sin echar de menos a alguno de ellos:

Requisito	Descripción
RU-01	El usuario puede crearse una cuenta
RU-02	El usuario puede iniciar sesión
RU-03	El usuario puede crear un viaje
RU-04	El usuario puede modificar los datos de un viaje
RU-05	El usuario puede eliminar un viaje
RU-06	El usuario puede crear una actividad dentro del viaje
RU-07	El usuario puede modificar los datos de una actividad
RU-08	El usuario puede eliminar una actividad
RU-09	El usuario puede guardar los datos de su equipaje
RU-10	El usuario puede modificar los datos del equipaje
RU-11	El usuario puede eliminar los datos de un equipaje
RU-12	El usuario puede almacenar los datos de un gasto
RU-13	El usuario puede modificar los datos de un gasto
RU-14	El usuario puede eliminar los datos de un gasto
RU-15	El usuario puede consultar los datos de un viaje
RU-16	El usuario puede consultar los datos de una actividad
RU-17	El usuario puede consultar los datos de su equipaje
RU-18	El usuario puede consultar los datos de un gasto
RU-19	El usuario puede modificar sus propios datos
RU-20	El usuario puede cerrar sesión
RU-21	El usuario puede editar sus datos

Tabla 26 - Requisitos de usuario del proyecto

4.3 Casos de uso

Un caso de uso es un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable. Los casos de uso proporcionan una estructura para expresar requisitos funcionales en el contexto de procesos empresariales y de sistema. Los casos de uso pueden representarse como un elemento gráfico en un diagrama y como una especificación de caso de uso en un documento textual.

Tras haber listado todos los requisitos de usuario, se hace más sencillo realizar el diagrama de casos de uso que el usuario puede llevar a cabo con la aplicación:

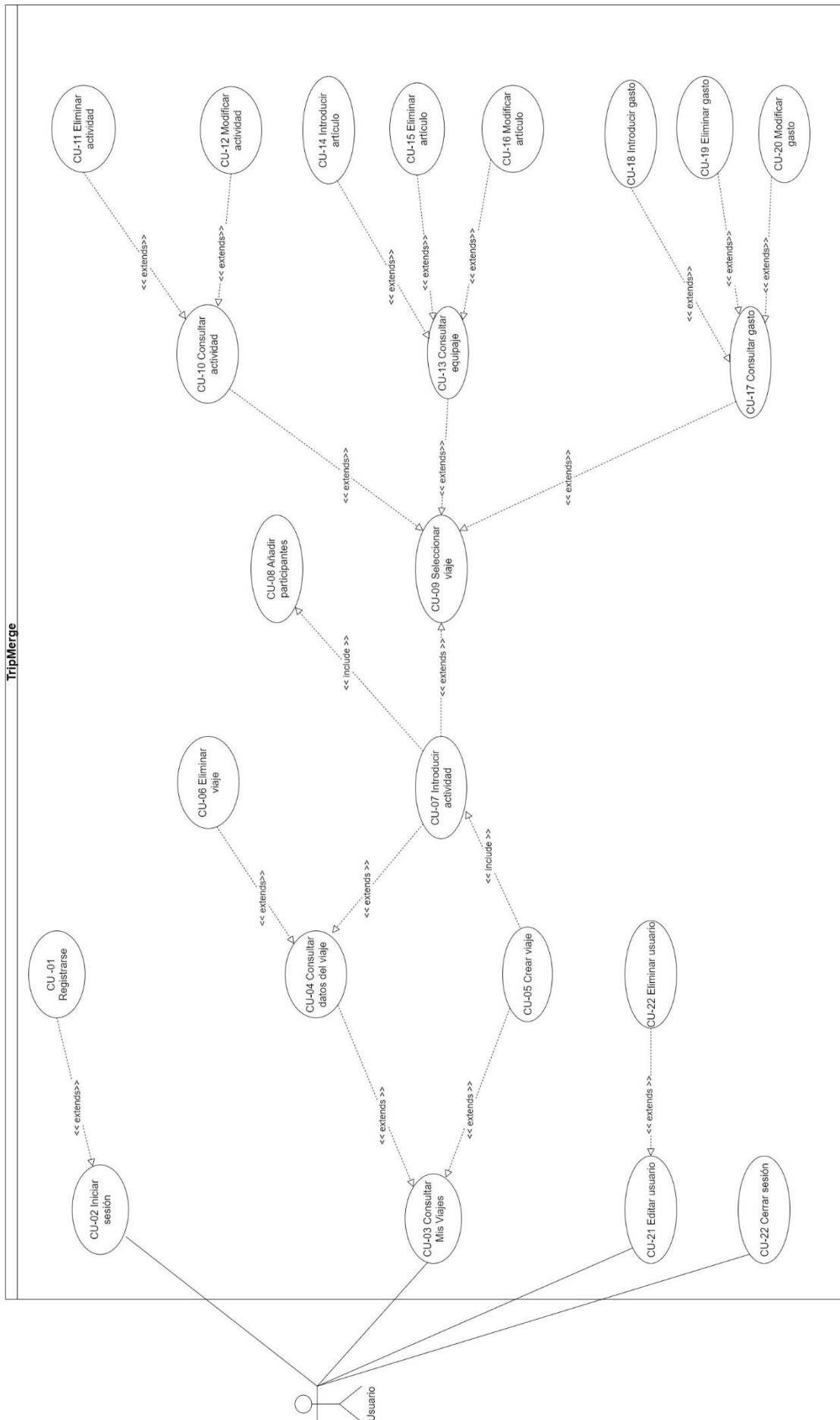


Ilustración 20 - Diagrama Casos de Uso

4.4 Especificación de casos de uso

CU-01	Registrarse
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	RU-01
Actor	Usuario
Descripción	El sistema permitirá al usuario crearse una cuenta.
Precondiciones	N/A
Flujo normal	<ol style="list-style-type: none"> 1. El usuario abre la aplicación. 2. El sistema muestra por pantalla un formulario. 3. El usuario introduce sus datos para crear la cuenta. 4. El sistema crea un nuevo usuario con los datos introducidos.
Postcondiciones	El usuario podrá iniciar sesión y utilizar las funciones de la aplicación.
Excepciones	<ol style="list-style-type: none"> 4.1 El usuario introduce los datos con un formato erróneo y aparece un mensaje de error. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal. 4.2 El usuario introduce un correo electrónico que ya existe en la base de datos y aparece un mensaje de error. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal.
Importancia	Alta

Tabla 27 - Especificación de CU-01

CU-02	Iniciar sesión
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-02 • RU-20
Actor	Usuario
Descripción	El sistema permitirá al usuario iniciar sesión con su cuenta.
Precondiciones	El usuario debe haberse creado una cuenta previamente.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario abre la aplicación. 2. El sistema muestra un formulario. 3. El usuario inicia sesión con el email y la contraseña introducidos. 4. El sistema comprueba que existe un usuario con esos datos e inicia sesión.
Postcondiciones	El sistema muestra la vista Home y permite al usuario utilizar las funciones de la aplicación.
Excepciones	<ol style="list-style-type: none"> 4.1 El usuario introduce los datos erróneos y aparece un mensaje de error. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal.
Importancia	Alta

Tabla 28 - Especificación de CU-02

CU-04	Consultar Mis Viajes
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-15
Actor	Usuario
Descripción	El sistema permitirá al usuario consultar todos sus viajes.
Precondiciones	El usuario debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Mis viajes’ en el menú de navegación. 2. El sistema muestra un listado con los viajes a los que pertenece el usuario.
Postcondiciones	El usuario visualiza los datos de los viajes.
Excepciones	<ol style="list-style-type: none"> 2.1 No haya datos ningún viaje creado. <ul style="list-style-type: none"> ○ Se mostraría un botón para crear un nuevo viaje.
Importancia	Media

Tabla 29 - Especificación de CU-03

CU-04	Consultar datos del viaje
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-15
Actor	Usuario
Descripción	El sistema permitirá al usuario consultar cualquier dato del viaje.
Precondiciones	El usuario debe haber iniciado sesión y haber creado un viaje.
Flujo normal	<ol style="list-style-type: none"> 3. El usuario pulsa el botón de ‘Mis viajes’ en el menú de navegación. 4. El sistema muestra un listado con los viajes a los que pertenece el usuario. 5. El usuario pulsa en el viaje el cual quiere visualizar los datos. 6. El sistema muestra los datos del viaje.
Postcondiciones	El usuario visualiza los datos del viaje elegido.
Excepciones	<ol style="list-style-type: none"> 2.2 No haya datos introducidos en el viaje. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal, indicando que no existe ningún dato.
Importancia	Media

Tabla 30 - Especificación de CU-04

CU-05	Crear viaje
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	RU-03
Actor	Usuario
Descripción	El sistema permitirá al usuario crear un viaje.
Precondiciones	El usuario debe haber iniciado sesión previamente.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Mis viajes’ en el menú de navegación. 2. El sistema muestra un listado de todos los viajes en los que participa el usuario. 3. El usuario pulsa el botón de ‘Crear viaje’. 4. El sistema muestra un formulario para los datos del viaje. 5. El usuario introduce todos los datos del viaje en cuestión. 6. El usuario pulsa el botón de crear viaje. 7. El sistema guarda todos los datos en la base de datos
Postcondiciones	El usuario podrá visualizar en cualquier momento los datos introducidos en ese viaje, ya sea equipaje, agenda, gastos, etc.
Excepciones	<ol style="list-style-type: none"> 7.1 Si los campos obligatorios no están completos se mostrará un mensaje de error y no se guardarán los datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 4 del flujo normal.
Importancia	Alta

Tabla 31 - Especificación de CU-05

CU-06	Eliminar viaje
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-05
Actor	Usuario
Descripción	El sistema permitirá al usuario eliminar cualquier viaje creado.
Precondiciones	El usuario debe haber iniciado sesión y haber creado un viaje
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Mis viajes’ en el menú de navegación. 2. El sistema muestra un listado con los viajes a los que pertenece el usuario. 3. El usuario pulsa en el viaje el cual quiere eliminar. 4. El sistema muestra los datos del viaje. 5. El usuario pulsa el botón de ‘Eliminar viaje’. 6. El sistema muestra un mensaje para confirmar la acción de eliminar. 7. El usuario confirma la petición. 8. El sistema elimina los datos y el viaje de la base de datos.
Postcondiciones	El usuario no podrá visualizar los datos de ese viaje ya que está eliminado de la base de datos. Deberá crear uno nuevo.
Excepciones	<ol style="list-style-type: none"> 6.1 Si el usuario no confirma la acción de eliminar la actividad no se eliminará de la base de datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal.
Importancia	Media

Tabla 32 - Especificación de CU-06

CU-07	Introducir actividad
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-06
Actor	Usuario
Descripción	El sistema permitirá al usuario crear una actividad a través de un viaje.
Precondiciones	El usuario debe haber iniciado sesión y haber creado un viaje.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de “Agenda” en el menú de navegación. 2. El sistema muestra la vista ‘Agenda’ en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje en el que quiere introducir la actividad. 4. El sistema muestra un calendario con todas las actividades que ya se han introducido. 5. El usuario pulsa el botón de ‘Añadir actividad’. 6. El sistema muestra un formulario para guardar los datos de la actividad. 7. El usuario introduce los datos. 8. El usuario pulsa el botón de ‘Guardar cambios’ 9. El sistema guarda todos los datos en la base de datos.
Postcondiciones	El usuario podrá consultar los datos de la actividad y crear más.
Excepciones	<ol style="list-style-type: none"> 9.1 Si los campos obligatorios no están completos se mostrará un mensaje de error y no se guardarán los datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 6 del flujo normal.
Importancia	Media

Tabla 33 - Especificación de CU-07

CU-08	Añadir participantes
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-06
Actor	Usuario
Descripción	El sistema permitirá al usuario añadir participantes a un viaje o actividad.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje o creado una actividad.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario se encuentra en el formulario de creación de un viaje o actividad. 2. El sistema muestra un campo para introducir el email de los participantes 3. El usuario inserta los emails de los usuarios que van a participar en el viaje. 4. El sistema el nombre y apellidos de los usuarios introducidos. 5. El usuario pulsa el botón de ‘Añadir viaje’ 6. El sistema almacena los datos en la base de datos.
Postcondiciones	N/A
Excepciones	<ol style="list-style-type: none"> 3.1 El email introducido no es correcto. - Se retomaría en el paso 2 del flujo normal. 3.2 Si se quiere introducir participantes en una actividad de un viaje, solo se pueden seleccionar los participantes que realizan la actividad anterior.
Importancia	Media

Tabla 34 - Especificación de CU-08

CU-09	Seleccionar viaje
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-06
Actor	Usuario
Descripción	El sistema permitirá al usuario seleccionar un viaje para poder consultar actividades, equipaje o gastos.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje o creado una actividad.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa ‘Equipaje’, ‘Gastos’ o ‘Agenda’ en el menú de navegación. 2. El sistema muestra un listado de todos los viajes del usuario. 3. El usuario pulsa el viaje del que quiere realizar alguna acción.
Postcondiciones	N/A
Excepciones	<ol style="list-style-type: none"> 1 No existe ningún viaje. - Se mostraría un botón para poder crear un nuevo viaje.
Importancia	Media

Tabla 35 - Especificación de CU-09

CU-10	Consultar actividad
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-06 • RU-16
Actor	Usuario
Descripción	El sistema permitirá al usuario consultar los datos de cualquier actividad creada.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje y creado una actividad.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de 'Agenda' en el menú de navegación. 2. El sistema muestra la vista 'Agenda' en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje en el que quiere consultar la actividad. 4. El sistema muestra un calendario con todas las actividades del viaje en cuestión. 5. El usuario pulsa la actividad la cual quiere consultar los datos. 6. El sistema muestra los datos de la actividad en cuestión.
Postcondiciones	El usuario visualiza los datos de la actividad seleccionada.
Excepciones	<ol style="list-style-type: none"> 1.1 El viaje o la actividad no hayan sido creados. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal.
Importancia	Media

Tabla 36 - Especificación de CU-10

CU-11	Eliminar actividad
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-06 • RU-08
Actor	Usuario
Descripción	El sistema permitirá al usuario eliminar cualquier actividad creada.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje y creado una actividad.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Agenda’ en el menú de navegación. 2. El sistema muestra la vista ‘Agenda’ en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje en el que quiere eliminar la actividad. 4. El sistema muestra un calendario con todas las actividades del viaje seleccionado. 5. El usuario pulsa en la actividad en cuestión. 6. El sistema muestra los datos de la actividad. 7. El usuario pulsa el botón de ‘Modificar datos’. 8. El sistema muestra un formulario para modificar la actividad. 9. El usuario pulsa el botón de ‘Eliminar actividad’. 10. El sistema muestra un mensaje para confirmar la acción de eliminar. 11. El usuario confirma la petición. 12. El sistema elimina los datos y la actividad de la base de datos.
Postcondiciones	El usuario no podrá visualizar ni interactuar con la actividad ya que ha sido eliminada.
Excepciones	<ol style="list-style-type: none"> 9.1 Si el usuario no confirma la acción de eliminar la actividad no se eliminará de la base de datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 6 del flujo normal.
Importancia	Media

Tabla 37 - Especificación de CU-11

CU-12	Modificar actividad
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-06 • RU-07
Actor	Usuario
Descripción	El sistema permitirá al usuario modificar los datos de cualquier actividad creada.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje y creado una actividad.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de 'Agenda' en el menú de navegación. 2. El sistema muestra la vista 'Agenda' en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje en el cual está la actividad a modificar. 4. El sistema muestra un calendario con todas las actividades del viaje seleccionado. 5. El usuario pulsa en la actividad en cuestión. 6. El sistema muestra los datos de la actividad. 7. El usuario pulsa en el botón de 'Modificar datos'. 8. El sistema muestra un formulario para modificar los datos. 9. El usuario rellena los campos de los datos de la actividad. 10. El usuario pulsa el botón 'Guardar cambios'. 11. El sistema modifica los datos de la actividad en la base de datos.
Postcondiciones	El usuario visualiza los datos de la actividad modificada.
Excepciones	<ol style="list-style-type: none"> 9.1 Si los campos obligatorios no están completos se mostrará un mensaje de error y no se guardarán los datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 9 del flujo normal.
Importancia	Media

Tabla 38 - Especificación de CU-12

CU-13	Consultar equipaje
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-09 • RU-10
Actor	Usuario
Descripción	El sistema permitirá al usuario consultar los datos del equipaje de un viaje.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje e introducido el equipaje.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de 'Equipaje' en el menú de navegación. 2. El sistema muestra la vista 'Equipaje' en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje el cual quiere consultar el equipaje. 4. El sistema muestra los datos del equipaje en cuestión. 5. El usuario visualiza los datos del equipaje.
Postcondiciones	El usuario visualiza los datos del equipaje seleccionado.
Excepciones	<ol style="list-style-type: none"> 3.1 El viaje o el equipaje no hayan sido creados. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal, indicando que no existe ningún equipaje creado.
Importancia	Media

Tabla 39 - Especificación de CU-13

CU-14	Introducir artículo
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-09
Actor	Usuario
Descripción	El sistema permitirá al usuario almacenar el equipaje de un viaje concreto.
Precondiciones	El usuario debe haber iniciado sesión y haber creado un viaje.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de 'Equipaje' en el menú de navegación. 2. El sistema muestra la vista 'Equipaje' en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje el cual quiere añadir el equipaje. 4. El sistema muestra los artículos del equipaje que ya se han introducido o un mensaje para informar que no hay ningún equipaje guardado. 5. El usuario pulsa el botón de 'Añadir artículo'. 6. El sistema muestra un formulario para guardar los datos del equipaje. 7. El usuario introduce los datos en los campos de la aplicación. 8. El usuario pulsa el botón de 'Guardar cambios'. 9. El sistema guarda todos los datos en la base de datos.
Postcondiciones	El usuario podrá visualizar, modificar o eliminar los datos introducidos del equipaje en cualquier momento.
Excepciones	N/A
Importancia	Media
Observaciones	N/A

Tabla 40 - Especificación de CU-14

CU-15	Eliminar artículo
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-09 • RU-11
Actor	Usuario
Descripción	El sistema permitirá al usuario eliminar cualquier equipaje.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje e introducido su equipaje.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de 'Equipaje' en el menú de navegación. 2. El sistema muestra la vista 'Equipaje' en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje el cual quiere eliminar el artículo. 4. El sistema muestra un listado de los artículos del equipaje del viaje seleccionado. 5. El usuario pulsa el icono de la papelera. 6. El sistema elimina el artículo de la base de datos.
Postcondiciones	El usuario no podrá visualizar ni interactuar con el artículo ya que ha sido eliminado.
Excepciones	N/A
Importancia	Media

Tabla 41 - Especificación de CU-15

CU-16	Modificar artículo
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-09 • RU-10
Actor	Usuario
Descripción	El sistema permitirá al usuario modificar los datos de cualquier equipaje introducido.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje e introducido su equipaje.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de 'Equipaje' en el menú de navegación. 2. El sistema muestra la vista 'Equipaje' en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje en el cual está el equipaje a modificar. 4. El sistema muestra un listado de todos los artículos del equipaje en el viaje seleccionado. 5. El usuario pulsa en el icono de las dos flechas. 6. El sistema cambia el estado de ese artículo (Adquirido o no adquirido).
Postcondiciones	El usuario visualiza el nuevo estado del artículo.
Excepciones	N/A
Importancia	Media

Tabla 42 - Especificación de CU-16

CU-17	Consultar gasto
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-12 • RU-18
Actor	Usuario
Descripción	El sistema permitirá al usuario consultar los datos del gasto de un viaje.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje e introducido los gastos.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Gastos’ en el menú de navegación. 2. El sistema muestra la vista ‘Gastos’ en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje el cual quiere consultar los gastos. 4. El sistema muestra un listado de los gastos. 5. El usuario visualiza los datos de los gastos.
Postcondiciones	El usuario visualiza los datos del equipaje seleccionado.
Excepciones	<ol style="list-style-type: none"> 2.1 El viaje o el gasto no hayan sido creados. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal, indicando que no existe ningún gasto.
Importancia	Media

Tabla 43 - Especificación de CU-17

CU-18	Introducir gasto
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-12
Actor	Usuario
Descripción	El sistema permitirá al usuario almacenar los gastos de un viaje concreto.
Precondiciones	El usuario debe haber iniciado sesión y haber creado un viaje.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Gastos’ en el menú de navegación. 2. El sistema muestra la vista ‘Gastos’ en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje el cual quiere añadir los gastos. 4. El sistema muestra los gastos que ya se han introducido o un mensaje para informar que no hay ningún gasto guardado. 5. El usuario pulsa el botón de ‘Añadir gasto’. 6. El sistema muestra un formulario para guardar los datos del gasto. 7. El usuario introduce los datos en los campos de la aplicación. 8. El usuario pulsa el botón de ‘Guardar gasto’. 9. El sistema guarda todos los datos en la base de datos.
Postcondiciones	El usuario podrá visualizar, modificar o eliminar los datos introducidos del gasto en cualquier momento.
Excepciones	<ol style="list-style-type: none"> 7.1 Si los campos obligatorios no están completos se mostrará un mensaje de error y no se guardarán los datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 6 del flujo normal.
Importancia	Media

Tabla 44 - Especificación de CU-18

CU-19	Eliminar gasto
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-12 • RU-14
Actor	Usuario
Descripción	El sistema permitirá al usuario eliminar cualquier gasto.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje e introducido su gasto.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Gastos’ en el menú de navegación. 2. El sistema muestra la vista ‘Gastos’ en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje el cual quiere eliminar el gasto. 4. El sistema muestra un listado de los gastos del viaje seleccionado. 5. El usuario pulsa el botón con el icono de un lápiz. 6. El sistema muestra un formulario para editar el gasto. 7. El usuario pulsa el botón ‘Eliminar gasto’. 8. El sistema muestra un mensaje para confirmar la acción de eliminar. 9. El usuario confirma la petición. 10. El sistema elimina los datos y el gasto de la base de datos.
Postcondiciones	El usuario no podrá visualizar ni interactuar con el gasto ya que ha sido eliminado.
Excepciones	<ol style="list-style-type: none"> 9.1 Si el usuario no confirma la acción de eliminar el gasto no se eliminará de la base de datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 4 del flujo normal.
Importancia	Media

Tabla 45 - Especificación de CU-16

CU-20	Modificar gasto
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-03 • RU-12 • RU-13
Actor	Usuario
Descripción	El sistema permitirá al usuario modificar los datos de cualquier gasto introducido.
Precondiciones	El usuario debe haber iniciado sesión, creado un viaje e introducido sus gastos.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ‘Gastos’ en el menú de navegación. 2. El sistema muestra la vista ‘Gastos’ en el que se muestra un listado de todos los viajes que tiene el usuario. 3. El usuario pulsa en el viaje el cual quiere modificar su gasto. 4. El sistema muestra un listado de todos los gastos en el viaje seleccionado. 5. El usuario pulsa en el botón con el icono de un lápiz. 6. El sistema muestra un formulario para modificar los datos. 7. El usuario rellena los campos de los datos del gasto. 8. El usuario pulsa el botón ‘Guardar cambios’. 9. El sistema modifica los datos de la actividad en la base de datos.
Postcondiciones	El usuario visualiza los datos del gasto modificado.
Excepciones	<ol style="list-style-type: none"> 7.1 Si los campos obligatorios no están completos se mostrará un mensaje de error y no se guardarán los datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 6 del flujo normal.
Importancia	Media

Tabla 46 - Especificación de CU-20

CU-21	Editar usuario
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-21
Actor	Usuario
Descripción	El sistema permitirá al usuario modificar sus datos excepto su email.
Precondiciones	El usuario debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa 'Editar usuario' en el desplegable del menú de navegación. 2. El sistema muestra un formulario con los datos actuales del usuario. 3. El usuario inserta los nuevos datos en el formulario. 4. El usuario pulsa el botón 'Guardar cambios'. 5. El sistema modifica los datos del usuario en la base de datos.
Postcondiciones	El usuario visualiza sus datos ya modificados.
Excepciones	<ol style="list-style-type: none"> 3.1 Si los campos obligatorios no están completos se mostrará un mensaje de error y no se guardarán los datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal.
Importancia	Media

Tabla 47 - Especificación de CU-21

CU-22	Eliminar usuario
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-21
Actor	Usuario
Descripción	El sistema permitirá al usuario eliminar por completo su cuenta de usuario.
Precondiciones	El usuario debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa 'Eliminar usuario' en el desplegable del menú de navegación. 2. El sistema muestra un formulario con los datos actuales del usuario. 3. El usuario pulsa el botón 'Eliminar usuario'. 4. El sistema muestra un mensaje para confirmar la acción de eliminar. 5. El usuario confirma la petición. 6. El sistema modifica los datos del usuario en la base de datos.
Postcondiciones	Se cierra sesión.
Excepciones	<ol style="list-style-type: none"> 5.1 Si el usuario no confirma la acción de eliminar el gasto no se eliminará de la base de datos. <ul style="list-style-type: none"> ○ Se retomaría en el paso 2 del flujo normal.
Importancia	Media

Tabla 48 - Especificación de CU-22

CU-23	Cerrar sesión
Versión	1.0
Autor	Rubén Velasco Plaza
Requisitos asociados	<ul style="list-style-type: none"> • RU-20
Actor	Usuario
Descripción	El sistema permitirá al usuario modificar sus datos excepto su email.
Precondiciones	El usuario debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario pulsa ‘Cerrar sesión’ en el desplegable del menú de navegación. 2. El sistema cierra sesión en la aplicación y redirige a la página de Login.
Postcondiciones	Se puede volver a iniciar sesión.
Excepciones	N/A
Importancia	Baja

Tabla 49 - Especificación de CU-23

4.5 Requisitos de información

Los requisitos de información son especificaciones detalladas que describen la información necesaria para cumplir con los objetivos de un sistema o proyecto. Estos requisitos son fundamentales en el desarrollo de sistemas de información, software o cualquier iniciativa que involucre la gestión y manipulación de datos.

A continuación, se mostrará un listado con los distintos requisitos de información del sistema:

Requisito	Descripción
RI-01	El sistema almacenará los datos de los usuarios que se hayan creado una cuenta.
RI-02	El sistema almacenará los datos de los viajes creados por los usuarios.
RI-03	El sistema almacenará los datos de las actividades introducidas por los usuarios.
RI-04	El sistema almacenará los datos de los equipajes introducidos en los viajes.
RI-05	El sistema almacenará los datos de los gastos producidos en los viajes.

Tabla 50 - Requisitos de información del proyecto

4.5.1 Diagrama Entidad-Relación

Para poder describir de una manera gráfica la información necesaria para este proyecto, se hará a través de un diagrama de entidad-relación con sus distintas componentes. Y, después, se describirán brevemente las entidades con sus correspondientes relaciones y en el siguiente apartado se centrará más en los tipos de atributos que contienen esas relaciones.

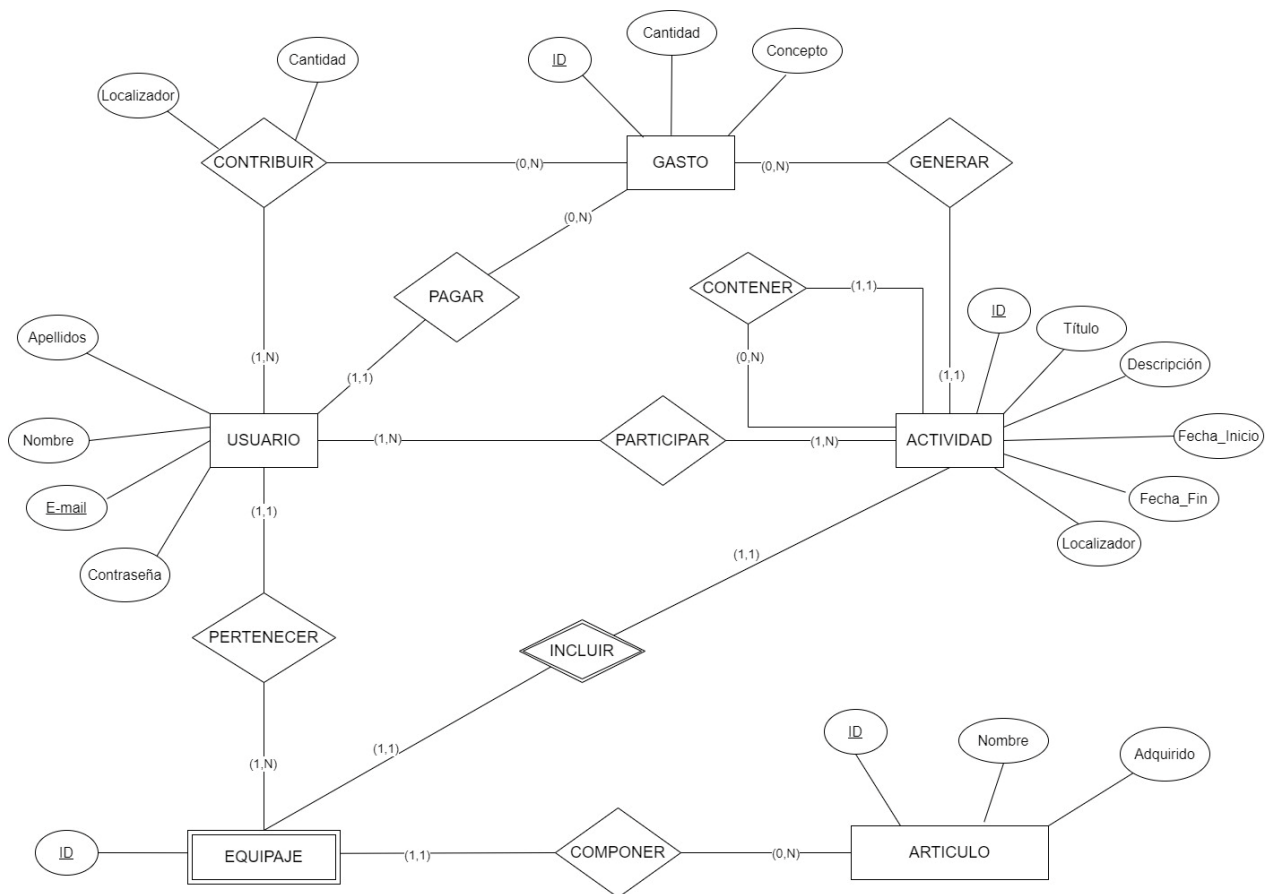


Ilustración 21 - Diagrama Entidad – Relación

Este proyecto, en la parte de la base de datos, se centra básicamente en 5 entidades principales:

- **Usuario:** En esta entidad se almacenará los datos de todos los usuarios que hayan creado una cuenta en la aplicación web. Para ello, deben introducir su nombre, apellidos, correo electrónico y contraseña.
- **Actividad:** Es la entidad que engloba todo el proyecto. Trata de almacenar los datos de los viajes creados por los usuarios, que, a través de ellos, se podrán realizar distintas acciones en la aplicación. Tiene como atributos un número de identificación, los participantes y la ubicación del destino.
- **Gasto:** Al realizar un viaje, es imposible escaparse de hacer un gasto cualquier día. Por lo tanto, esta entidad almacenará esos datos para llevar a cabo una gestión al finalizar el trayecto, como son un número de identificación, la cantidad y un nombre para reconocerlo posteriormente.
- **Equipaje:** Cada usuario dispondrá de un equipaje donde se almacenan todos los artículos. Dicho equipaje cambiará dependiendo del viaje que se elija.

- **Artículo:** El sistema almacenará los datos de los artículos que se quieren introducir en el equipaje del viaje. Para ello se guardará el nombre del artículo y si está adquirido o no. Esta característica existe para tener un control del equipaje días antes de realizar el viaje o de preparar la vuelta a casa.

Una vez descritas todas las relaciones del diagrama Entidad-Relación, se procede a explicar las relaciones que unen una entidad con otra:

- **Participar:** Se encarga de relacionar los usuarios con las distintas actividades. Se llega a la conclusión que un usuario tiene una o varias actividades y, viceversa, una actividad puede realizarse por uno o varios usuarios.
- **Contener:** Se trata de una relación recursiva para la entidad de actividad. La primera observación fue introducir una relación para los viajes y otra para las actividades de dentro del viaje. Pero, finalmente, a un viaje principal se le considera también una actividad. Para llevar eso a cabo, es necesario la relación de ‘Contener’ para unir los viajes con sus actividades secundarias.
- **Generar:** En cada viaje/actividad es inevitable producir algún gasto que otro. Por lo tanto, se unen con la relación de ‘Generar’. En este caso, una actividad genera ningún o muchos gastos y un gasto solo es generado por una actividad. Por lo que, se considera “padre” a la entidad ‘Actividad’ e “hijo” a la entidad ‘Gasto’.
- **Contribuir:** Dichos gastos son generados por los usuarios de la aplicación, por lo tanto, se unen por la relación ‘Contribuir’. En esta relación se encuentran dos atributos:
 - **Localizador:** Se trata de un *String* que identifica a la actividad en cuestión. Se compone de los números de identificación de las actividades contiguas, separadas cada una por un guion. Ejemplo: 15-24-31 (El gasto es producido en la actividad 31, la cual está dentro de la actividad 24 que a su vez se encuentra en el viaje 15).
 - **Cantidad:** Informa del total de dinero acumulado en ese gasto.
- **Pagar:** Un gasto debe ser pagado si o si por un usuario, por lo tanto, esta relación proporciona un atributo para almacenar dicho valor. Como gasto es la entidad hija en este caso, se le otorga el atributo ‘email_pagador’.
- **Pertenecer:** Cada usuario lleva consigo un equipaje y estas dos entidades están relacionadas con ‘Pertenecer’. Se puede observar en la cardinalidad que un usuario puede tener ningún o muchos equipajes, pero un equipaje solo es generado por un usuario.
- **Incluir:** Dicho equipaje se relaciona también con la actividad ya que dependiendo de la actividad se utiliza un equipaje u otro. Tras ello, una actividad posee cero o varios equipajes, pero un equipaje solo se lleva a una actividad.

- **Componer:** Para finalizar, esta relación se encarga de unir el equipaje total con cada uno de los artículos depositados en él. Por lo tanto, un equipaje podrá llevar cero o muchos artículos mientras que un artículo sólo es llevado en un solo equipaje.

4.5.2 Diccionario de datos

Para ser más precisos, se va a analizar entidad por entidad en el siguiente diccionario de datos para obtener más información sobre ellas, sus atributos y las relaciones. A continuación, se muestran varias tablas con los datos previamente comentados:

Entidad 'Usuario'				
Atributo	Tipo	Único	Puede ser nulo	Observaciones
E-mail	<i>String</i>	Sí	No	
Nombre	<i>String</i>	No	No	
Apellidos	<i>String</i>	No	No	
Contraseña	<i>String</i>	No	No	Se almacena mediante un hash con Crypto AES.

Tabla 51 - Entidad Usuario

Entidad 'Actividad'				
Atributo	Tipo	Único	Puede ser nulo	Observaciones
ID	<i>Integer</i>	Sí	No	
Título	<i>String</i>	No	No	
Descripción	<i>String</i>	No	No	
Fecha Inicio	<i>Date</i>	No	No	
Fecha Fin	<i>Date</i>	No	No	
Localizador	<i>String</i>	No	No	Se almacena concatenando las ids de cada actividad.

Tabla 52 - Entidad Actividad

Entidad 'Equipaje'				
Atributo	Tipo	Único	Puede ser nulo	Observaciones
ID	<i>Integer</i>	Sí	No	

Tabla 53 - Entidad Equipaje

Entidad 'Artículo'				
Atributo	Tipo	Único	Puede ser nulo	Observaciones
ID	<i>Integer</i>	Sí	No	
Adquirido	<i>Boolean</i>	No	No	Describe si el usuario tiene consigo dicho artículo o no
Nombre	<i>String</i>	No	No	

Tabla 54 - Entidad Artículo

Entidad 'Gasto'				
Atributo	Tipo	Único	Puede ser nulo	Observaciones
ID	<i>Integer</i>	Sí	No	
Concepto	<i>String</i>	No	No	Título o breve descripción del gasto
Cantidad	<i>Float</i>	No	No	

Tabla 55 - Entidad Gasto

Capítulo 5: DISEÑO

5.1 Arquitectura lógica

La arquitectura lógica se ocupa de la organización de los componentes de software, sus funciones y cómo se comunican entre sí para cumplir con los requisitos del sistema. También puede abordar cuestiones como la gestión de datos, el flujo de información, la lógica de negocio y las relaciones entre los diversos módulos o capas del software.

Al trabajar con una aplicación de escritorio, la arquitectura lógica se divide en tres capas significativas que se describen a continuación:

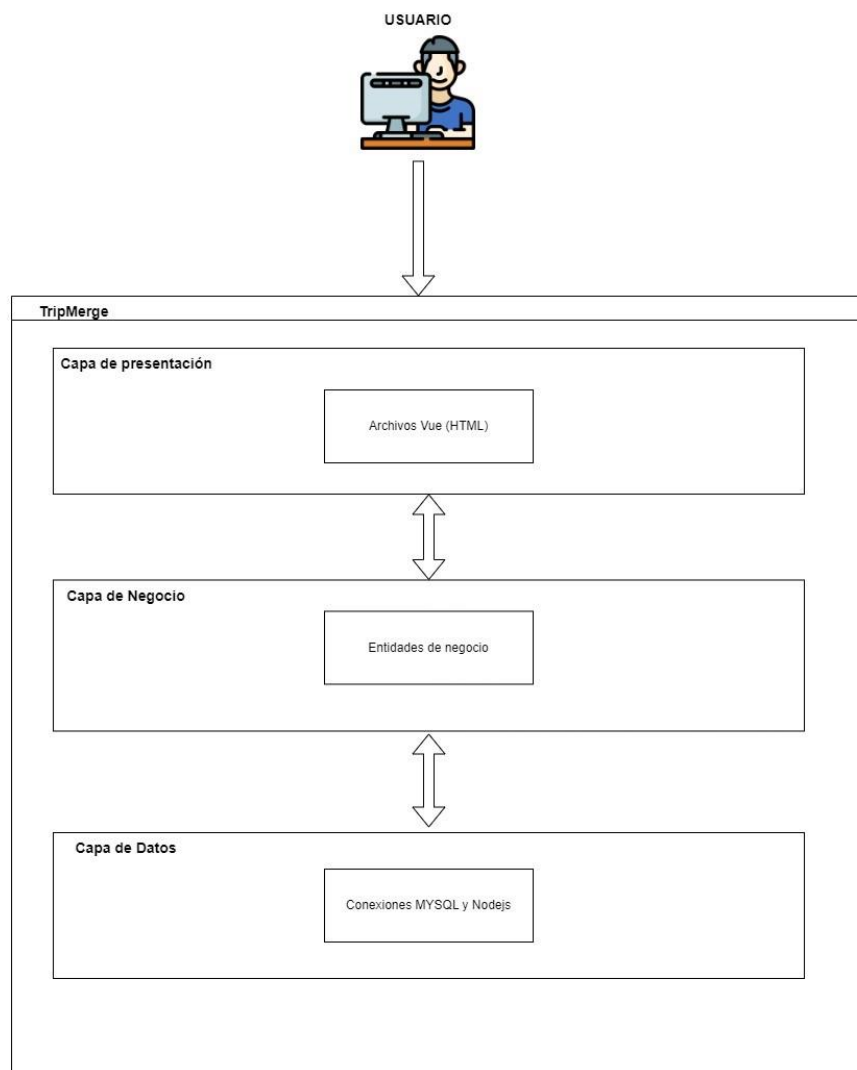


Ilustración 22 - Diagrama de la Arquitectura Lógica

1. Capa de presentación

En esta primera capa, se encuentran todas las interfaces de la aplicación, desde el login o registro hasta la última funcionalidad del sistema. En otras palabras, es la parte con la que el usuario puede interactuar de una manera sencilla.

2. Capa de negocio

En esta segunda capa, se encuentra toda la lógica de negocio que se encarga de unir la información entregada por el usuario y las respuestas de la base de datos.

3. Capa de datos

En esta última capa, se recogen las consultas entregadas por la lógica de negocio, a través del usuario, y se transforman en datos de la base de datos del sistema.

La aplicación web de este proyecto está diseñada mediante el patrón Modelo Vista Controlador (MVC) ya que se considera la mejor dada sus características. En la siguiente figura se procede a representar sus principales componentes:

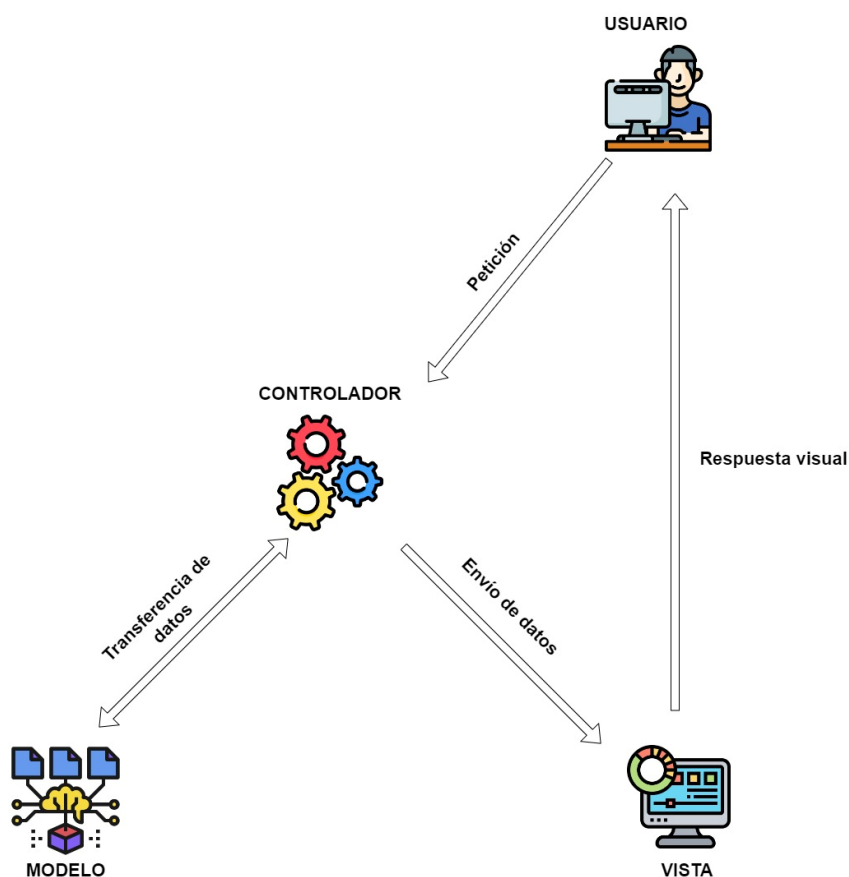


Ilustración 23 - Patrón Modelo Vista Controlador

- El usuario se comunica con la aplicación mediante la interfaz de usuario mandando distintas peticiones al controlador.
- El controlador gestiona las peticiones recibas intercambiando datos con el modelo.
- Tras ese intercambio, el controlador envía los datos proporcionados por el modelo a la vista.
- Finalmente, la vista realiza una respuesta visual a través de esa interfaz para que el usuario pueda observar el resultado.

5.2 Arquitectura física

La arquitectura física de un programa se refiere a la estructura y disposición física de los componentes de hardware que ejecutan el software. En el contexto de la programación, la arquitectura física se relaciona con cómo se distribuyen y organizan los recursos de hardware para ejecutar un programa o sistema de software.

Para documentar la arquitectura física del proyecto, se muestra primero el diagrama siguiente y después se describe brevemente:

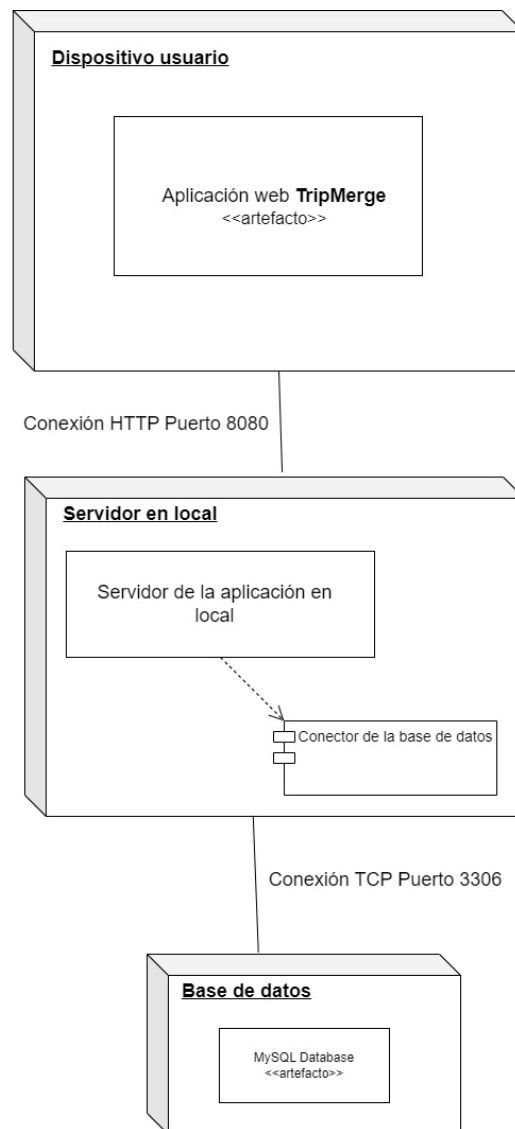


Ilustración 24 - Diagrama de despliegue

- 1. Dispositivo usuario:** Cliente para dar soporte a la aplicación de escritorio del proyecto. Su principal objetivo es enseñar la información que necesita el usuario para poder utilizar la aplicación. Se conecta con el servidor con el protocolo HTTP y por el puerto 8080.
- 2. Servidor en local:** Proporciona el servidor en el cual se encuentra la aplicación que se utiliza para dar un punto de unión entre las interfaces y la base de datos. Para la conexión con la base de datos se utiliza el protocolo TCP por el puerto 3306.
- 3. Base de datos:** Es la base de datos de MySQL, encargada de almacenar todos los datos necesarios para el usuario que esté utilizando la aplicación.

5.3 Diseño del modelo de datos

En este apartado, se diseña el modelo de datos utilizado para la base de datos, a partir del diagrama de Entidad-Relación, incluido en el Apartado 4.5.1.

Dicho modelo se obtiene teniendo en cuenta las relaciones que conectan las entidades y su cardinalidad.

USUARIO (Email, Contraseña, Nombre, Apellidos)

PARTICIPAR (Email_Usuario, ID_Actividad)

FK: Email_Usuario -> **USUARIO**(Email)

FK: ID_Actividad -> **ACTIVIDAD**(ID)

ACTIVIDAD (ID, Título, Descripción, Fecha_Inicio, Fecha_Fin, Localizador, ID_Viaje)

FK: ID_Viaje -> **ACTIVIDAD**(ID)

GASTO (ID, Cantidad, Concepto, ID_Actividad, email_pagador)

FK: ID_Actividad -> **ACTIVIDAD**(ID)

FK: email_pagador -> **USUARIO** (email)

CONTRIBUIR (Email_Usuario, ID_Gasto, Localizador, Cantidad)

FK: Email_Usuario -> **USUARIO**(Email)

FK: ID_Gasto -> **GASTO**(ID)

EQUIPAJE (ID, id_actividad, email)

FK: ID_Actividad -> **ACTIVIDAD**(ID)

FK: email -> **USUARIO** (email)

ARTICULO (ID, Nombre, Adquirido, ID_Equipaje)

FK: ID_Equipaje -> **EQUIPAJE**(ID)

Todas las *foreign keys* introducidas en el diseño lógico se introducen con la actualización y eliminación en cascada para facilitar el manejo de la base de datos, haciendo así esas acciones automáticas.

Dependiendo de las cardinalidades de las relaciones que unen dos entidades, se debe añadir o no atributos extra para el diseño de la base de datos. Los ejemplos de estos atributos son:

- Id_viaje en ACTIVIDAD
- Id_actividad en GASTO
- Id_actividad en EQUIPAJE
- Email en EQUIPAJE
- Id_equipaje en ARTÍCULO

Se debe puesto que, en el caso que las cardinalidades son 1 a N, se considera como entidad padre (cardinalidad 1) y entidad hijo (cardinalidad N). En dicho caso, la entidad hijo recibe un atributo extra que se corresponde a la *primary key* de la entidad padre. Ejemplo:

La entidad Actividad es padre de Gasto, con lo cual Gasto recibe `id_actividad` como *foreign key*.

Existe otro caso, el cual se percibe como las cardinalidades N a N, el cual crea una tabla adicional con dos atributos, las dos *primary keys* de las dos entidades relacionadas. Ejemplo:

La tabla Participar que relaciona Usuario con Actividad y recibe el email del usuario y la id de la actividad.

5.4 Diagrama de clases

En esta sección, se representa el diagrama de clases utilizado en la aplicación web de este proyecto. El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los otros objetos. A continuación, se muestra el diagrama:

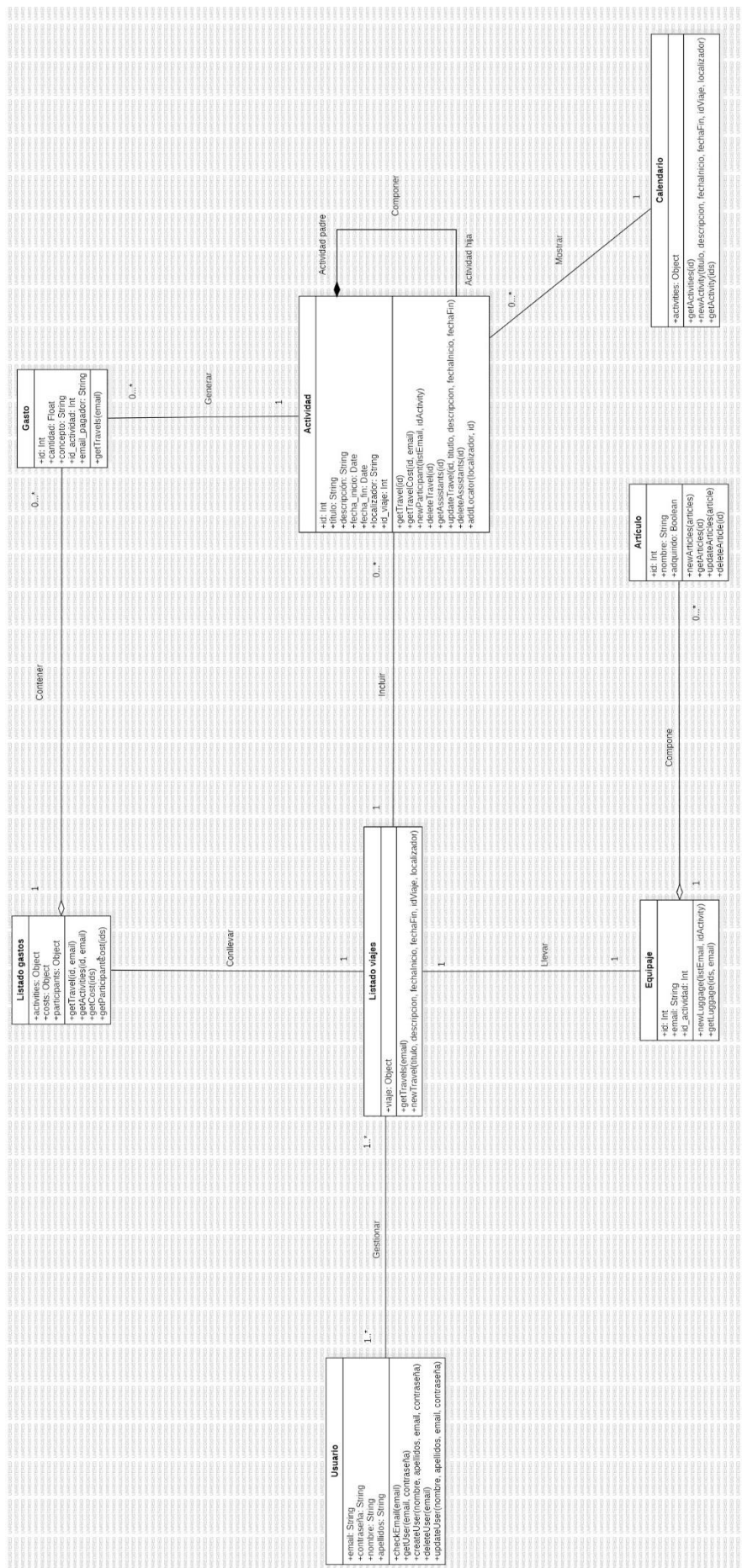


Ilustración 25 - Diagrama de clases

5.5 Diagrama de secuencia

Un diagrama de secuencia es una representación visual que muestra la interacción entre objetos en un sistema a lo largo del tiempo. Este tipo de diagrama es parte de los diagramas de comportamiento en UML (*Unified Modeling Language*) y se utiliza comúnmente en el diseño y análisis de sistemas de software.

En un diagrama de secuencia, se representan las interacciones entre diferentes objetos o componentes del sistema en forma de una secuencia temporal. Los objetos se muestran como cajas con líneas de vida verticales, y las interacciones entre ellos se muestran como flechas que indican los mensajes que se envían entre los objetos.

En este caso, se mostrarán los dos diagramas de secuencia necesarios para realizar el caso de uso de creación de un viaje. Primero, el usuario debe iniciar sesión en la aplicación para poder acceder a ella:

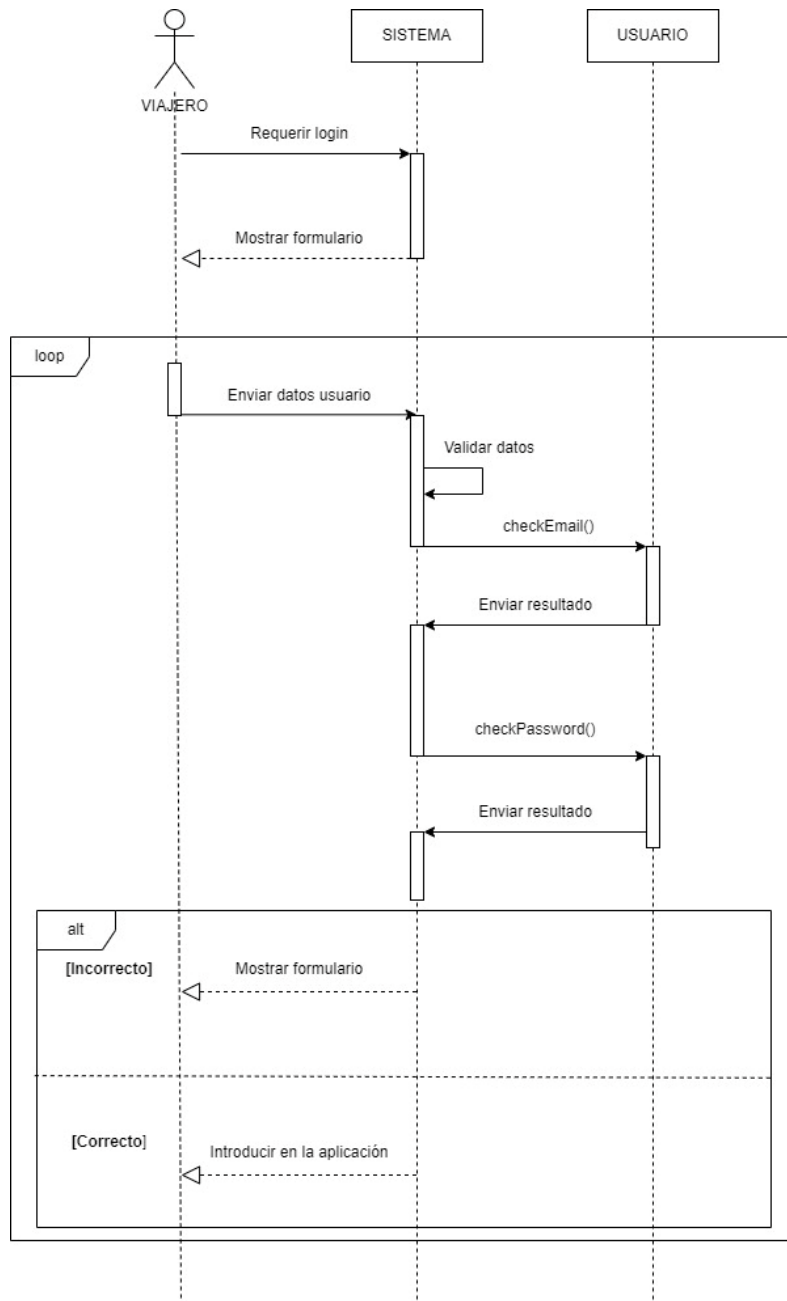


Ilustración 26 - Diagrama de secuencia de login

Y, finalmente, el usuario podrá añadir los viajes que quiera de la siguiente manera:

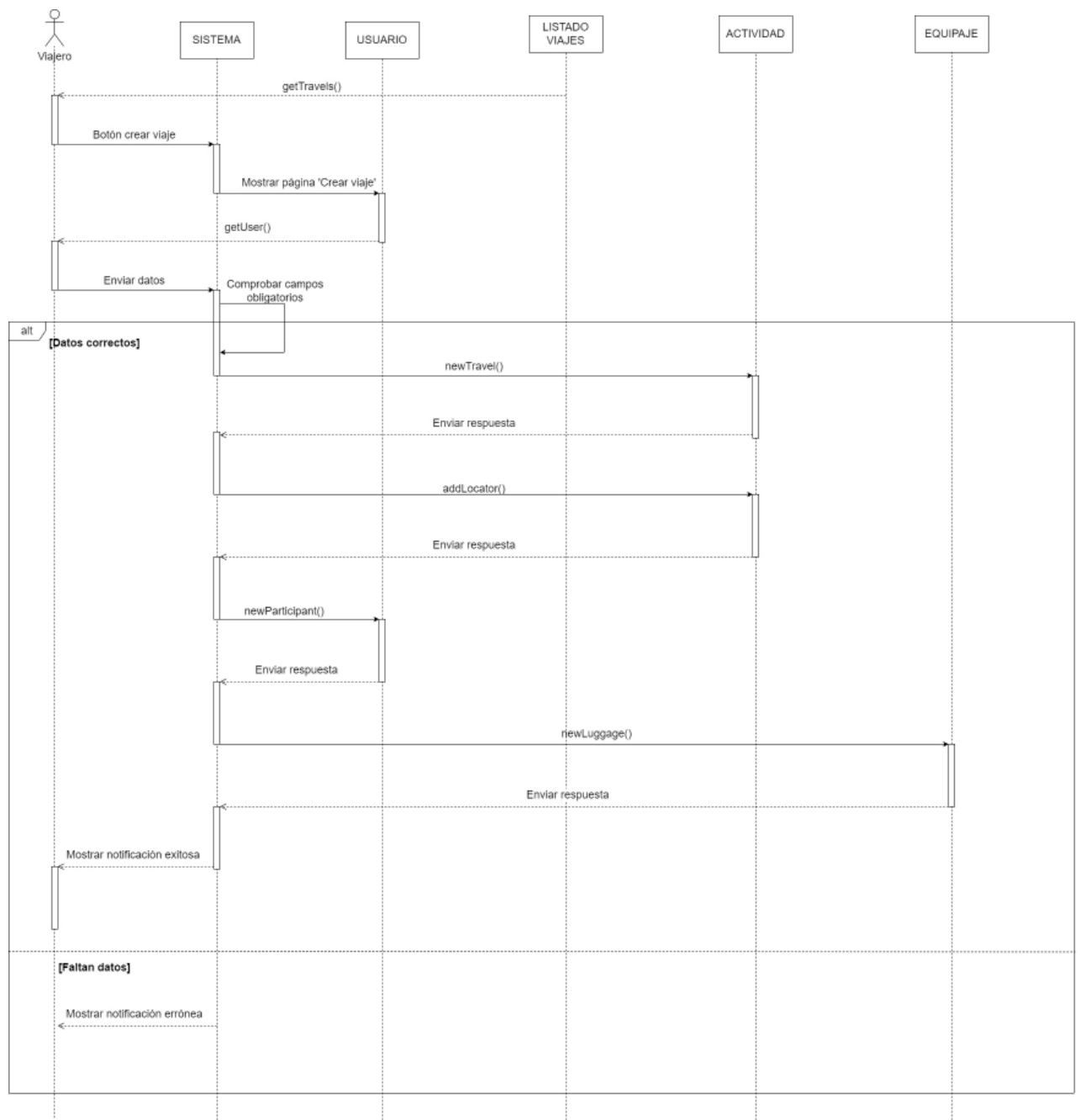


Ilustración 27 - Diagrama de secuencia creación viaje

5.6 Diseño de interfaz de usuario

En esta sección, se mostrarán los diferentes diseños de cada una de las vistas de la aplicación. Corresponden con las primeras ideas con lo que pueden variar con las interfaces finales. Todos los diseños han sido realizados con el programa BALSAMIQ WIREFRAMES.

Login

La primera vista nada más arrancar la aplicación corresponde con el Login. Consta de dos campos de texto para introducir el email y la contraseña del usuario previamente creado.

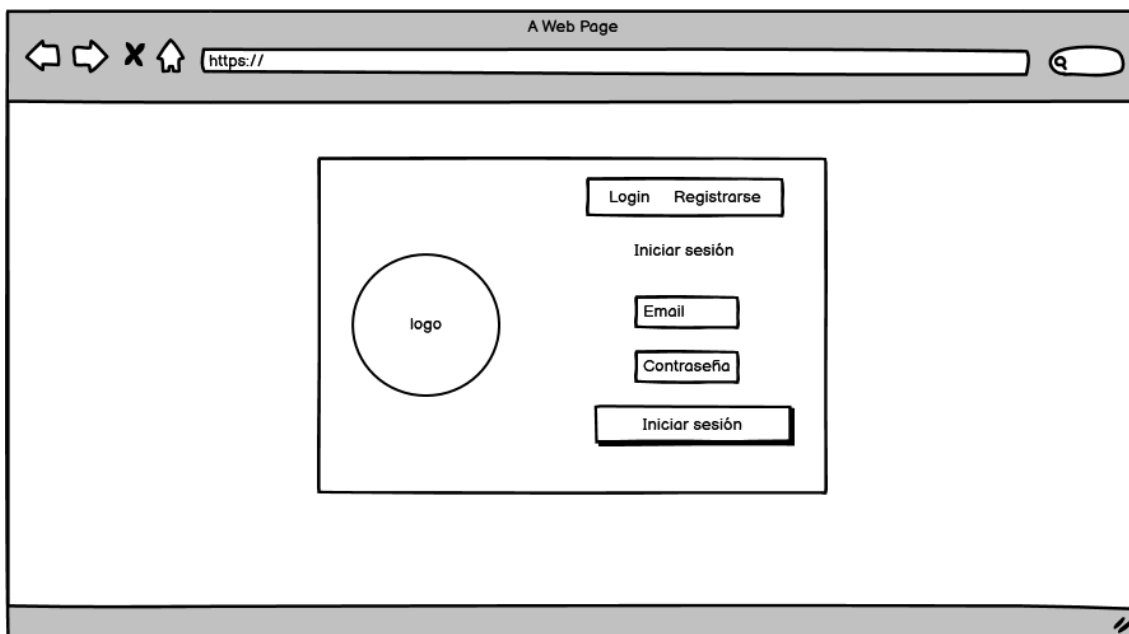


Ilustración 28 - Interfaz login

Registro

Si es la primera vez que el usuario utiliza la aplicación, es necesario crearse una cuenta. Por lo tanto, se crea esta interfaz con cuatro campos de texto para introducir los distintos datos de la persona. Se cuenta con el nombre, apellidos, email y contraseña.

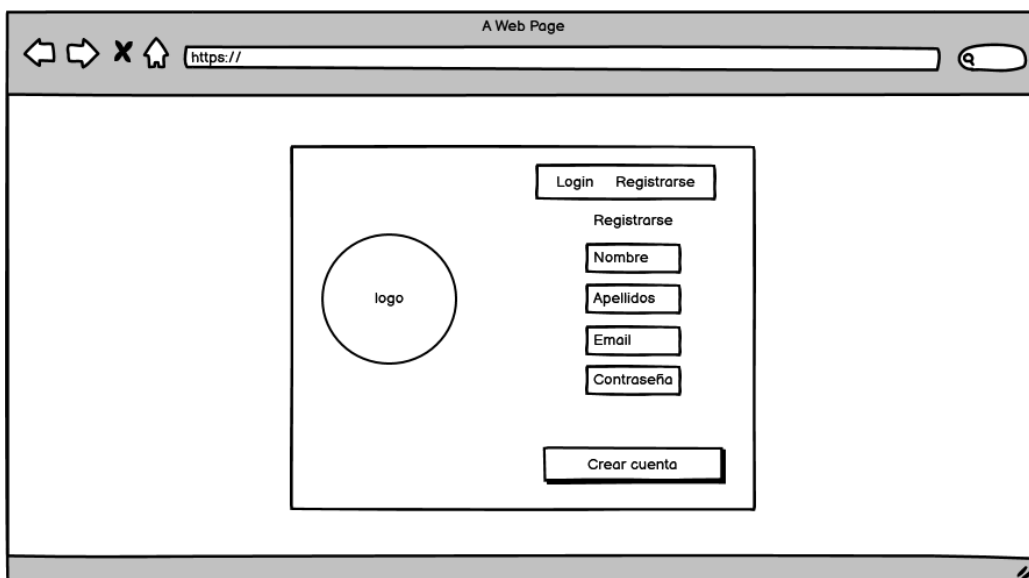


Ilustración 29 - Interfaz registro

Mis Viajes

Una vez iniciado sesión, el usuario se encontrará con la vista de los viajes totales, dónde se encontrará con toda la información de cada viaje. También podrá navegar con los botones para observar los datos de la agenda, equipaje y gastos. Se inserta también un buscador para filtrar los viajes y un botón para crear uno nuevo.



Ilustración 30 - Interfaz mis viajes

Consulta de los datos del viaje

Al hacer clic sobre un viaje, se abrirá un modal con los datos de dicho viaje. En este modal, habrá dos botones, uno de ellos para eliminar ese viaje y otro para modificar los datos.

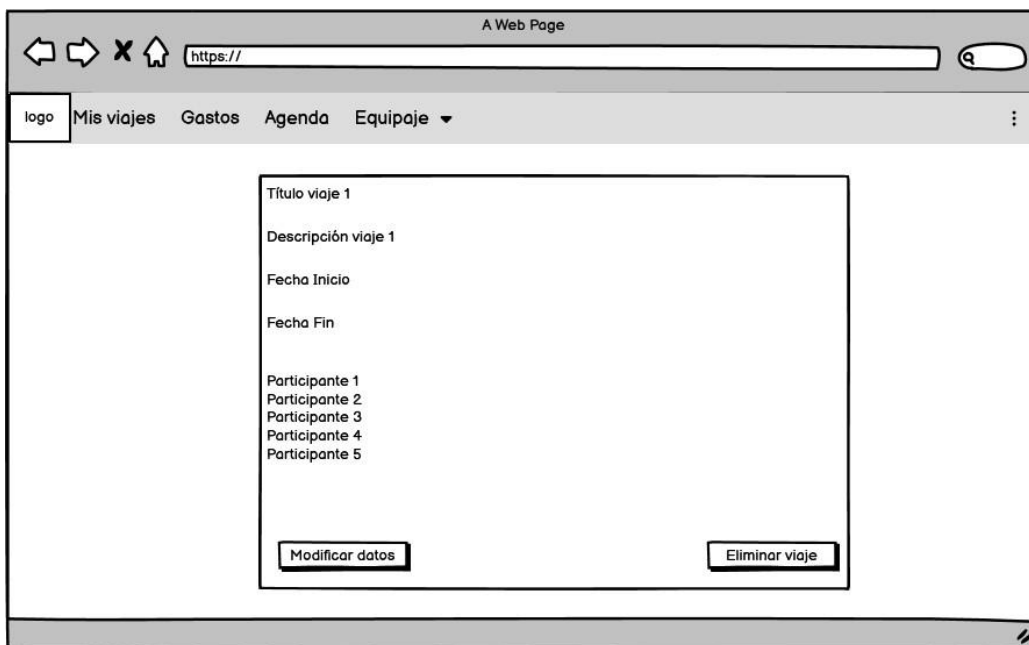


Ilustración 31 - Interfaz datos de un viaje

Creación / Modificación de los datos del viaje

Una vez el usuario haya hecho clic en el anterior botón de Modificar los datos o de añadir un viaje, se navegará hacia esta vista. Aquí se podrán añadir o cambiar los datos previamente introducidos por otros distintos. Tras haber hecho los cambios pertinentes, el usuario podrá guardarlos o eliminar el viaje por completo gracias a los dos botones situados en la parte inferior de la pantalla.

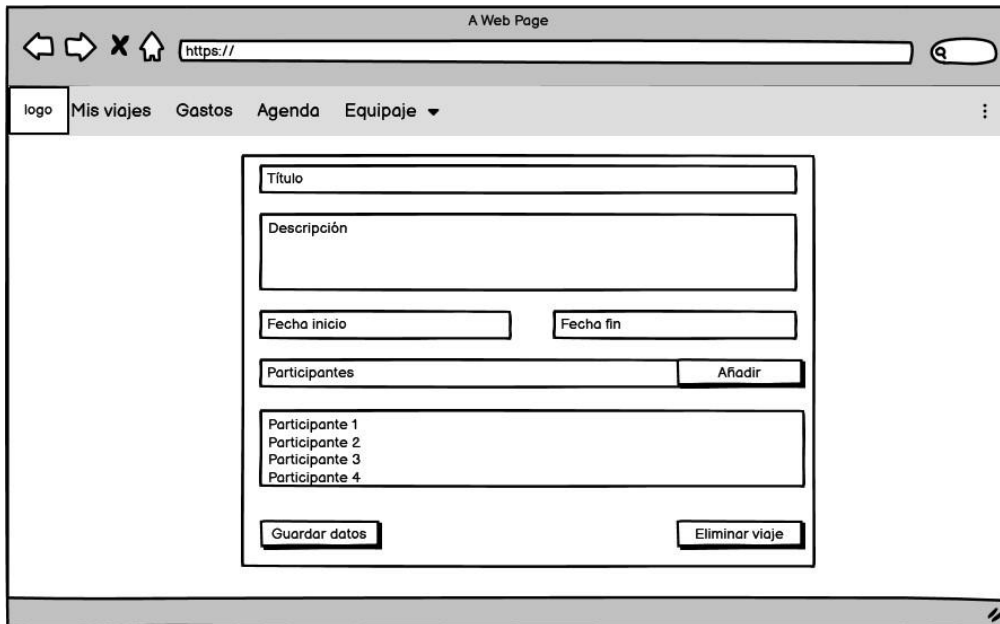


Ilustración 32 - Interfaz creación o modificación viaje

Gastos

En el apartado de los gastos, se mostrará una tabla con el listado de todos los viajes del usuario para que pueda elegir del que quiere ver sus gastos. Dicha tabla mostrará el título, descripción, fechas y un botón para acceder a él. También se podrá filtrar por título.



Ilustración 33 - Interfaz gastos

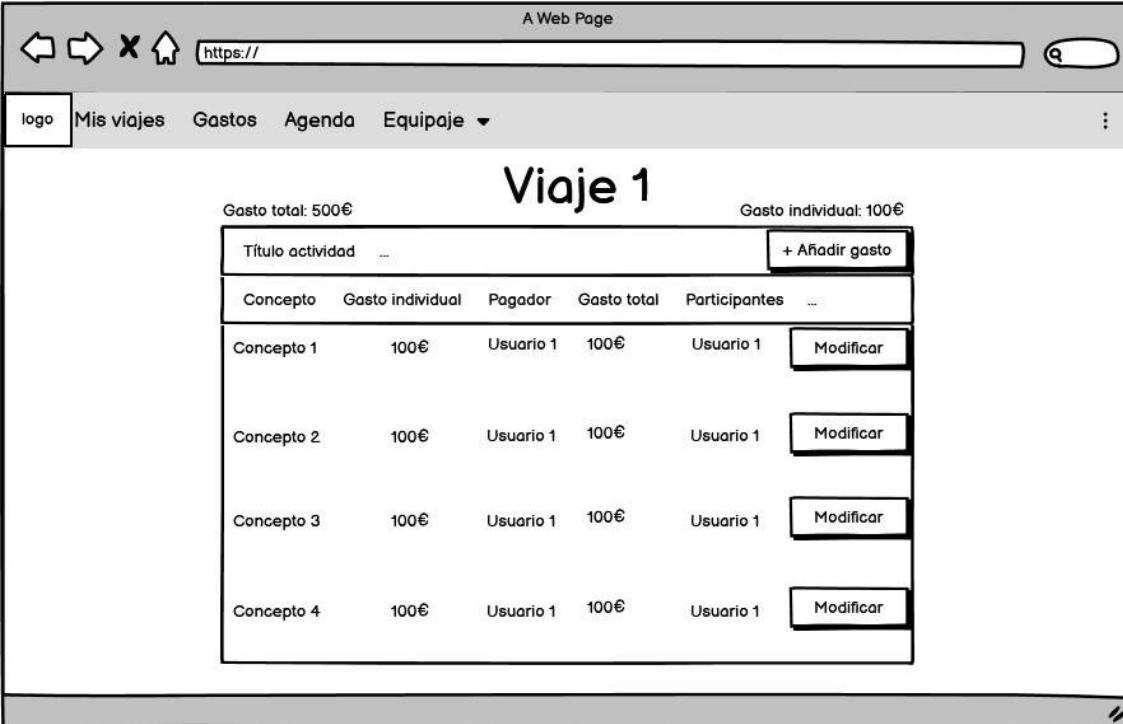
Datos de un gasto

Al pulsar el anterior botón se mostrarán varias tablas, cada una de una actividad distinta.

Antes de las tablas se imprimirá por pantalla el gasto total que tiene el viaje y el gasto individual total del usuario.

En cada tabla se podrá observar todos los datos de los gastos de la actividad en cuestión:

- Concepto
- Gasto individual
- Pagador
- Gasto total
- Participantes
- Botón para añadir gasto
- Botón para modificar el gasto



The screenshot shows a web browser window with the URL 'https://'. The page title is 'A Web Page'. The navigation menu includes 'logo', 'Mis viajes', 'Gastos', 'Agenda', and 'Equipaje'. The main content area is titled 'Viaje 1' and displays the following information:

Gasto total: 500€ Gasto individual: 100€

Título actividad ...					+ Añadir gasto
Concepto	Gasto individual	Pagador	Gasto total	Participantes	...
Concepto 1	100€	Usuario 1	100€	Usuario 1	Modificar
Concepto 2	100€	Usuario 1	100€	Usuario 1	Modificar
Concepto 3	100€	Usuario 1	100€	Usuario 1	Modificar
Concepto 4	100€	Usuario 1	100€	Usuario 1	Modificar

Ilustración 34 - Interfaz datos de un gasto

Creación / Modificación de los datos de un gasto

Una vez pulsado el botón anterior de modificar o de añadir gasto, se mostrará otra vista con dos campos de texto para el concepto y la cantidad, un selector para elegir el pagador del gasto y unas *checkbox* para los participantes del gasto.

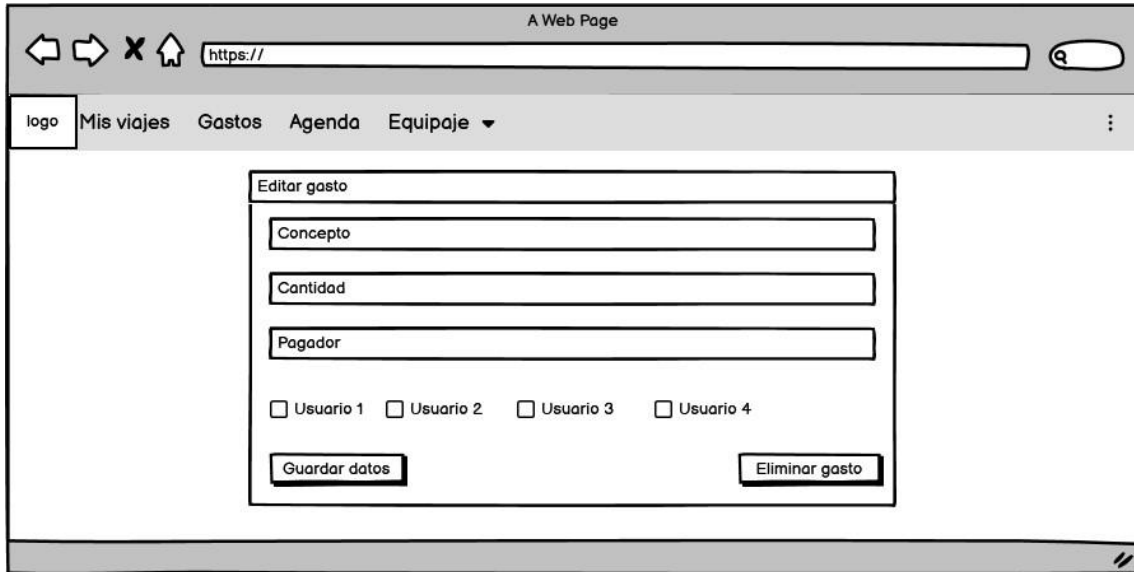


Ilustración 35 - Interfaz creación o modificación gasto

Agenda

En el apartado de la agenda se mostrará también una tabla con el listado de todos los viajes.

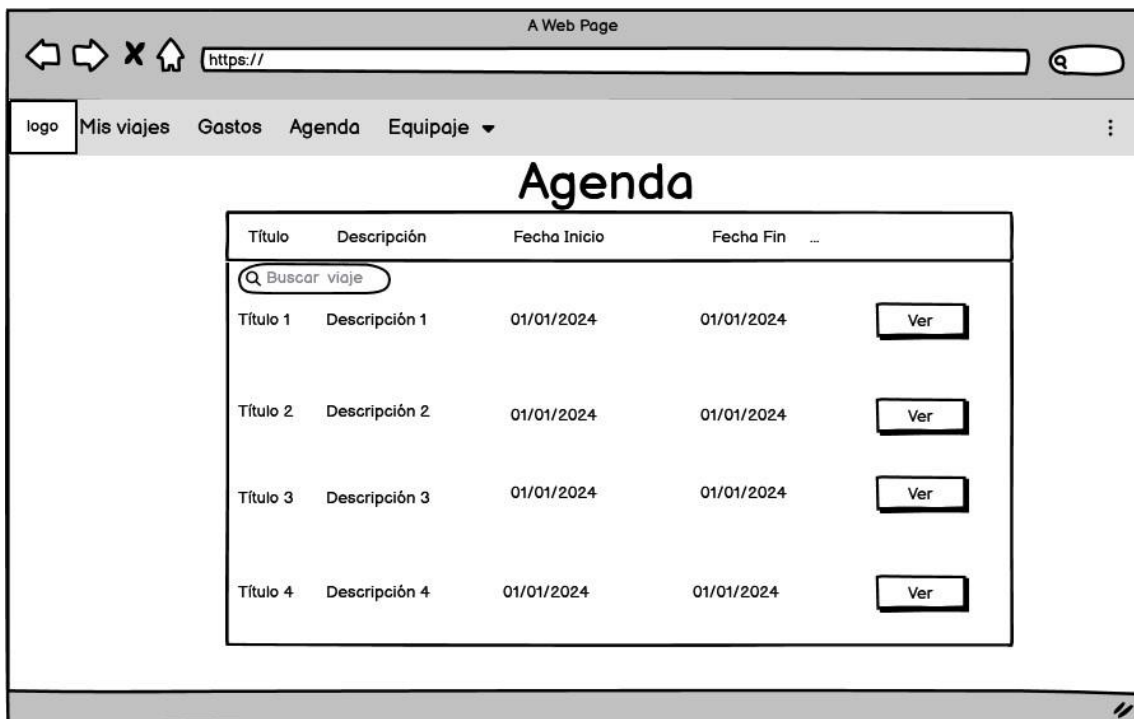


Ilustración 36 - Interfaz agenda

Calendario de las actividades

Al pulsar el botón de un viaje anterior, se imprimirá por pantalla un calendario de gran tamaño para poder ver todas las actividades que tiene el viaje principal. Cada una de ellas se verá reflejada en sus días con su título para diferenciarlas. También se podrán añadir más actividades gracias al botón de arriba a la derecha.

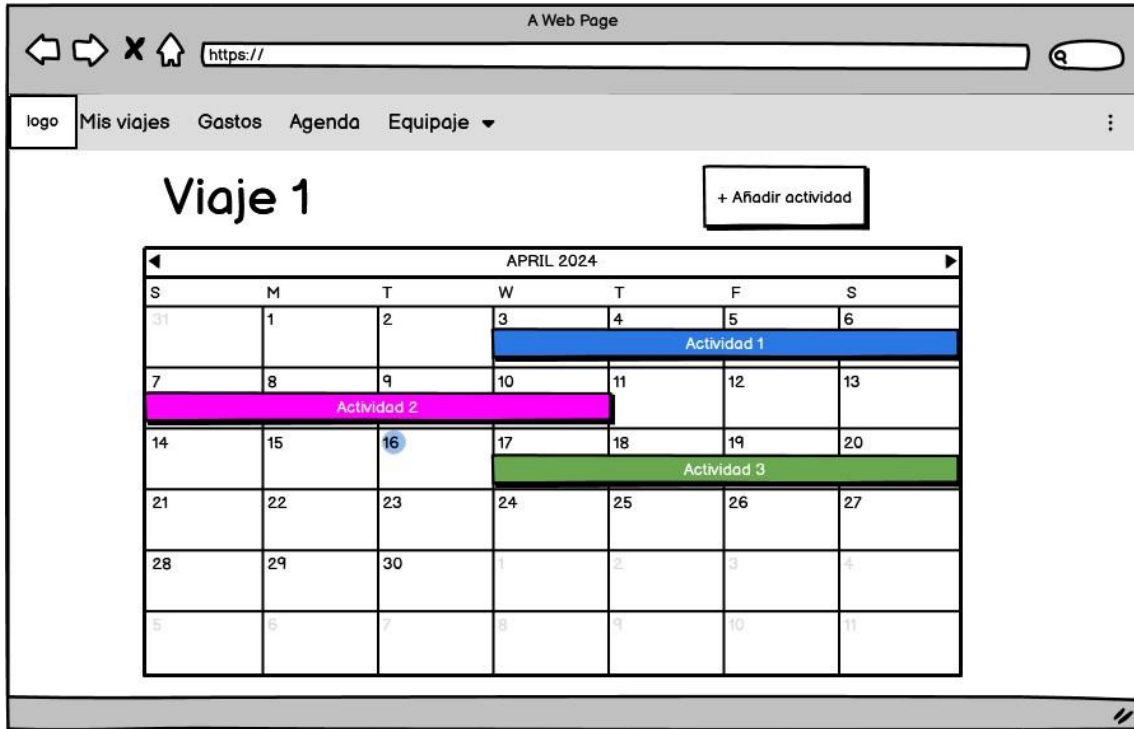


Ilustración 37 - Interfaz calendario

Consulta de los datos de una actividad

Al hacer *click* sobre una actividad del calendario, se abrirá un modal en el cual se podrá ver todos sus datos. También se podrá añadir una nueva actividad dentro de ella y modificar sus datos.

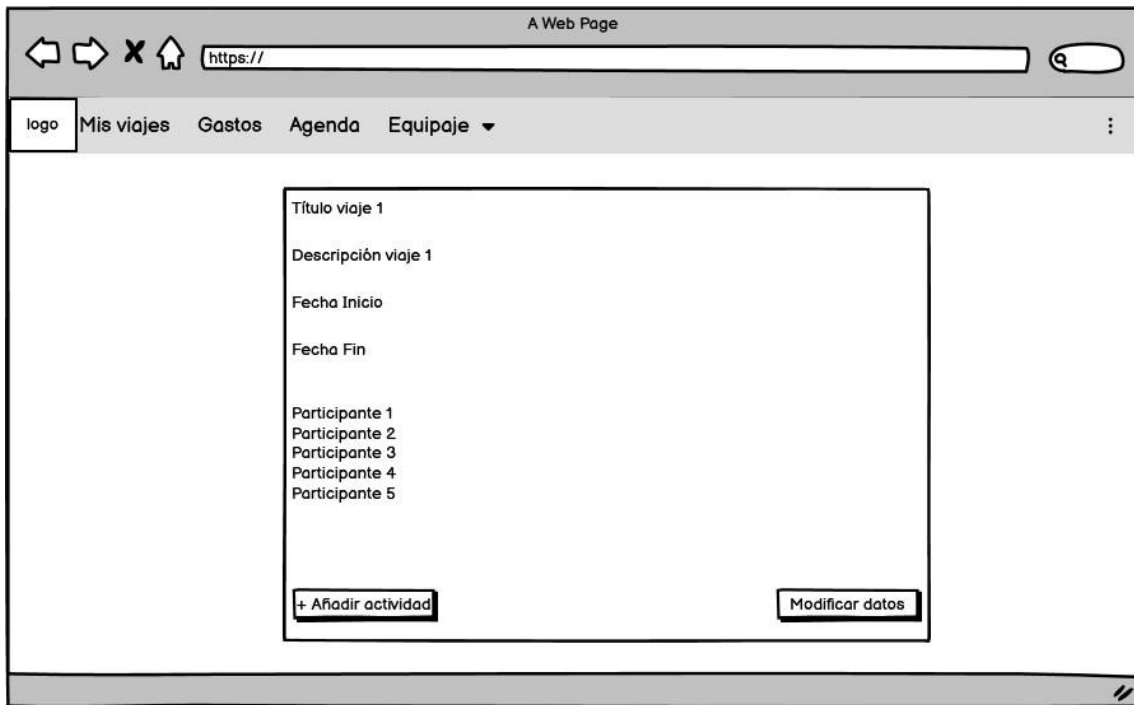


Ilustración 38 - Consulta actividad

Creación / Modificación de una actividad

Al pulsar el botón de añadir o modificar actividad se mostrarán unos campos de texto para cada dato de la actividad y unas *checkbox* para elegir los participantes de ella.

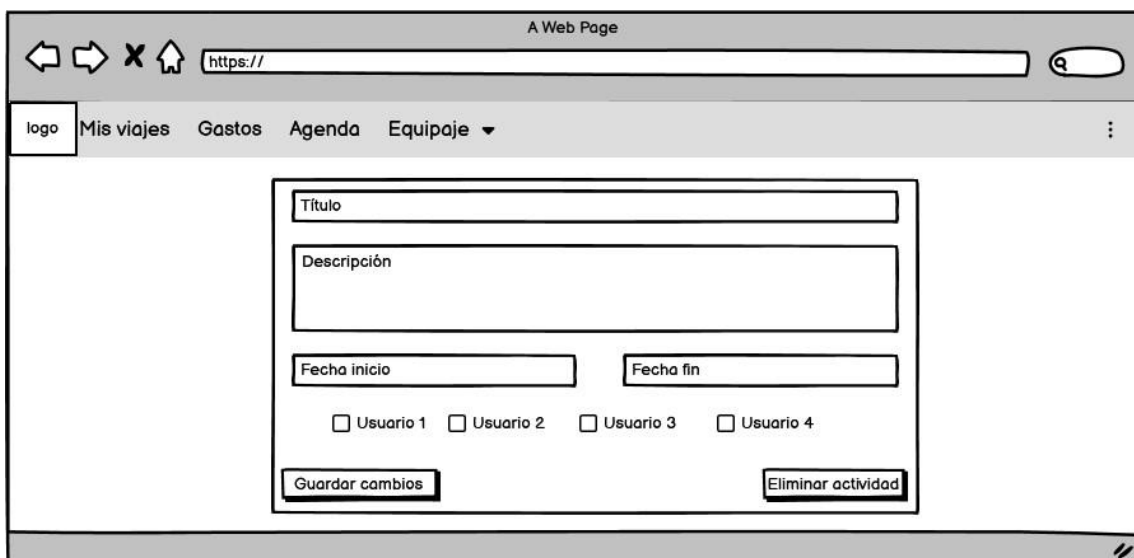


Ilustración 39 - Interfaz creación o modificación de actividad

Equipaje

En el apartado del equipaje, se podrá observar una tabla con el listado de todos los viajes del usuario.

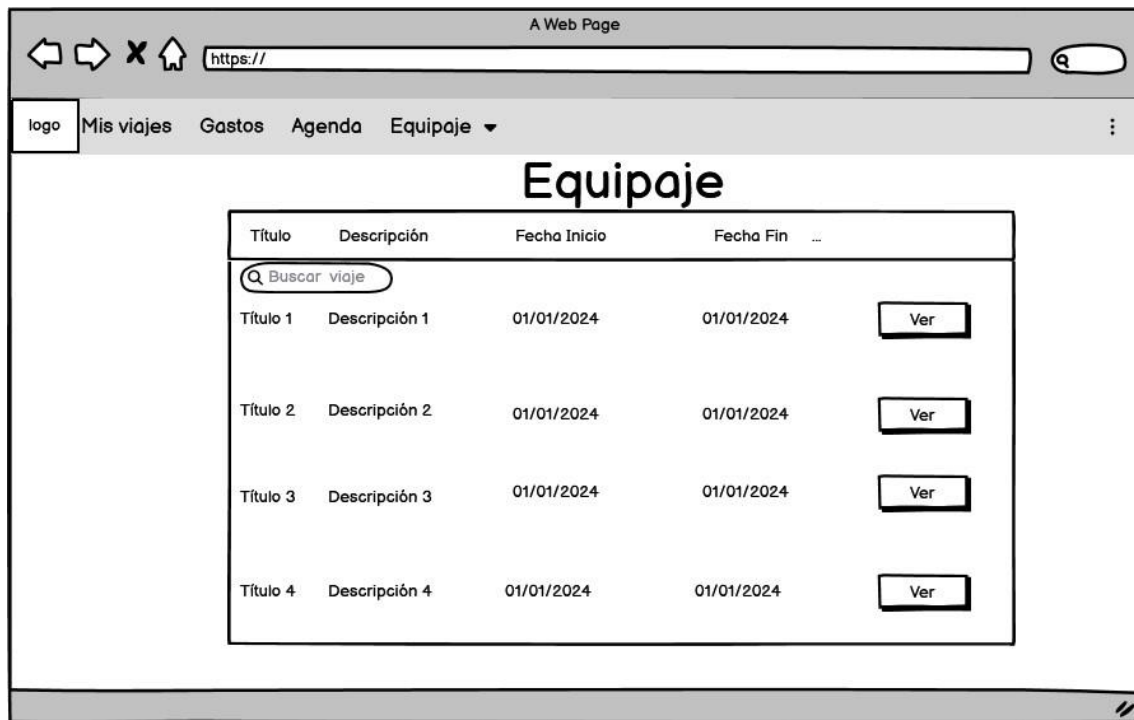


Ilustración 40 - Interfaz equipaje

Artículos del equipaje

Al elegir un viaje, se abrirá un modal con una barra de progreso que indica el número total de artículos que el usuario ya ha adquirido. A continuación, se mostrarán dos listados:

- Adquiridos: Artículos que el usuario posee
- No adquiridos: Artículos que el usuario todavía no tiene consigo

Finalmente, un botón para poder cerrar ese modal y otro botón para poder añadir más artículos.

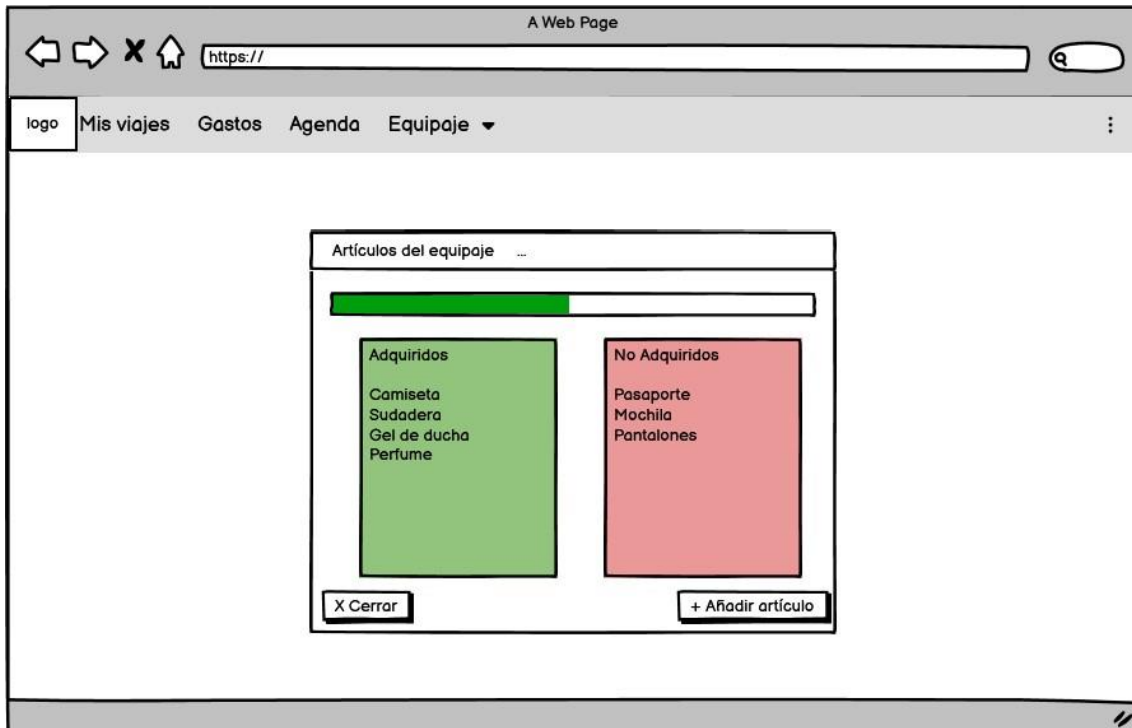
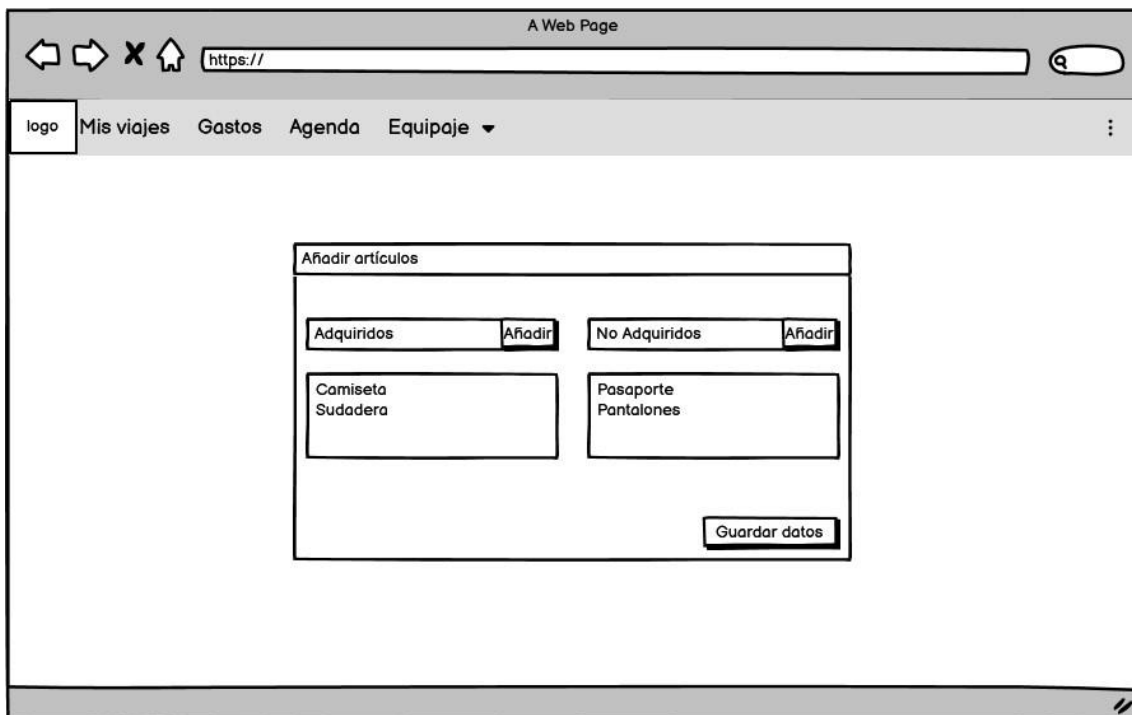


Ilustración 41 - Interfaz artículos

Añadir artículos

Una vez pulsado el botón de añadir artículo, el usuario podrá añadir los artículos pertinentes en su equipaje mediante dos campos de texto, uno para los adquiridos y otro para los no adquiridos. Podrá guardarlos mediante el botón de abajo a la derecha.

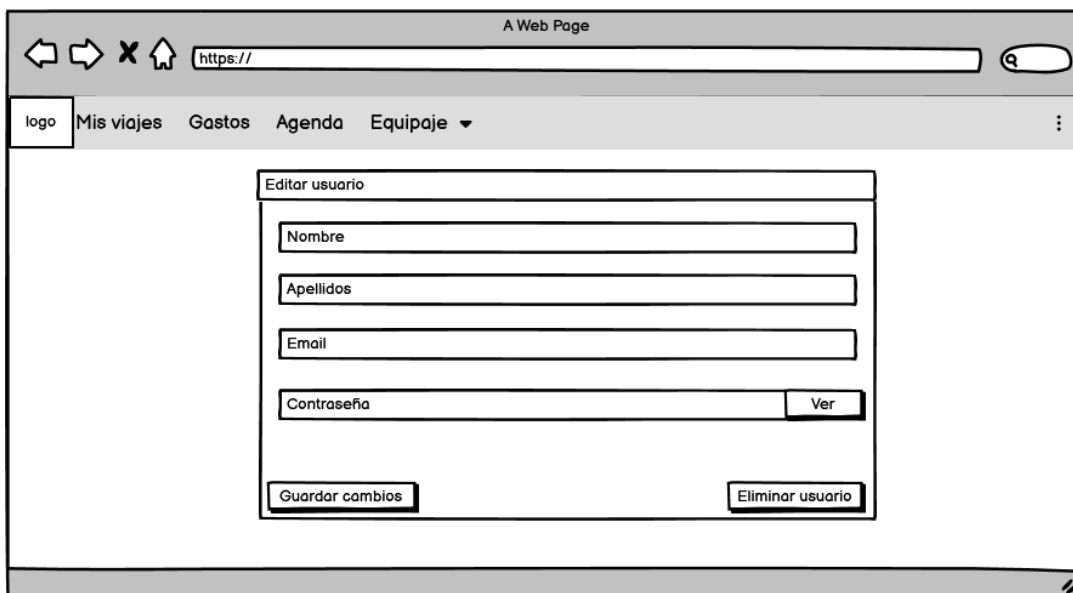


The screenshot shows a web browser window titled 'A Web Page'. The address bar contains 'https://'. The navigation menu includes 'logo', 'Mis viajes', 'Gastos', 'Agenda', and 'Equipaje'. The main content area displays a form titled 'Añadir artículos'. The form has two columns: 'Adquiridos' and 'No Adquiridos'. Each column has an 'Añadir' button. Below the 'Adquiridos' column, there are two text input fields containing 'Camiseta' and 'Sudadera'. Below the 'No Adquiridos' column, there are two text input fields containing 'Pasaporte' and 'Pantalones'. At the bottom right of the form is a 'Guardar datos' button.

Ilustración 42 - Interfaz añadir artículos

Editar usuario

Finalmente, el usuario mediante el desplegable de la derecha en el menú de navegación podrá editar sus datos mediante los siguientes campos de texto:



The screenshot shows a web browser window titled 'A Web Page'. The address bar contains 'https://'. The navigation menu includes 'logo', 'Mis viajes', 'Gastos', 'Agenda', and 'Equipaje'. The main content area displays a form titled 'Editar usuario'. The form has four text input fields: 'Nombre', 'Apellidos', 'Email', and 'Contraseña'. The 'Contraseña' field has a 'Ver' button next to it. At the bottom left of the form is a 'Guardar cambios' button, and at the bottom right is an 'Eliminar usuario' button.

Ilustración 43 - Interfaz editar usuario

Capítulo 6: IMPLEMENTACIÓN

En este capítulo se describirán algunas cuestiones relacionadas y reseñables sobre la implementación de la aplicación web. Para ello, se dividirá el capítulo en varios apartados.

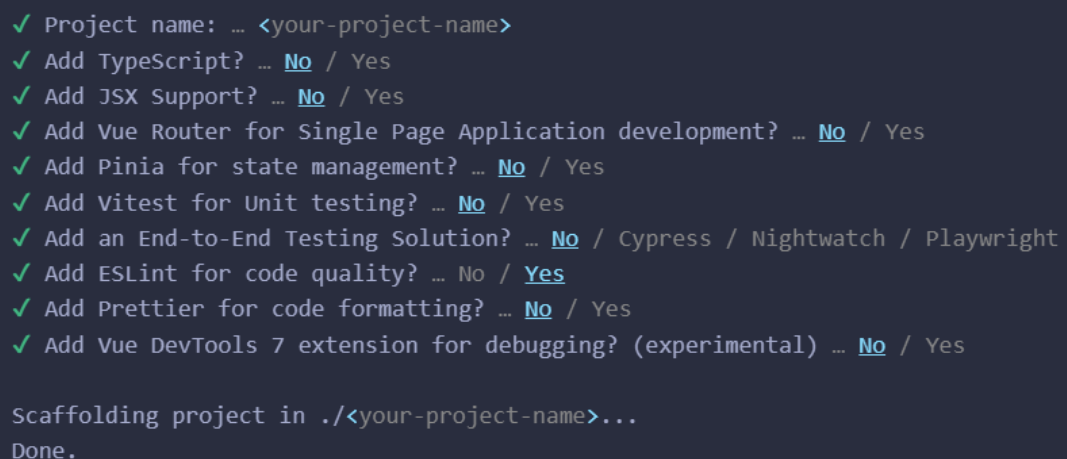
6.1 Guía rápida para desplegar aplicaciones basadas en Vue.js

Para conseguir este proyecto ha sido necesario crear dos proyectos en Visual Studio Code, uno de ellos para la parte de *front-end* y el otro para *back-end*. Al utilizar Vue para la realización de esta aplicación, se han seguido sus propios pasos para la creación de los proyectos. Dichos pasos se encuentran documentados en su página web oficial y son públicos para todos los usuarios. Se procede a explicarlos brevemente:

1. En primer lugar, se ha de obtener un editor de código fuente y el servidor Node.js.
2. Después, en la terminal se dirige hacia la ruta en la cual se ha de almacenar el código del proyecto (por ejemplo, C:/Documentos/proyecto) y se introduce el siguiente comando:

```
$ cd C:/Documentos/proyecto
$ npm create vue@latest
```

3. A continuación, aparecerán las siguientes opciones y se ha de seleccionar las marcadas en azul:



```
✓ Project name: ... <your-project-name>
✓ Add TypeScript? ... No / Yes
✓ Add JSX Support? ... No / Yes
✓ Add Vue Router for Single Page Application development? ... No / Yes
✓ Add Pinia for state management? ... No / Yes
✓ Add Vitest for Unit testing? ... No / Yes
✓ Add an End-to-End Testing Solution? ... No / Cypress / Nightwatch / Playwright
✓ Add ESLint for code quality? ... No / Yes
✓ Add Prettier for code formatting? ... No / Yes
✓ Add Vue DevTools 7 extension for debugging? (experimental) ... No / Yes

Scaffolding project in ./<your-project-name>...
Done.
```

Ilustración 44 - Opciones de creación del proyecto

4. Finalmente, se vuelven a introducir otros tres comandos
 - a. Uno para dirigirse al proyecto ya creado.
 - b. Otro para instalar las librerías y *plugins* incluidos en el fichero package.json.
 - c. Y el último para arrancar la plantilla de la aplicación:

```
$ cd <your-project-name>
$ npm install
$ npm run dev
```

Ilustración 45 - Comandos de creación del proyecto

Tras haber realizado los pasos anteriores para los dos proyectos, la manera de organizar los ficheros es totalmente diferente, al tratarse de proyectos de distinta naturaleza el *front-end* y el *back-end*.

Vue sigue el patrón de diseño MVC (Modelo, Vista, Controlador), de modo que en el *back-end* se despliega el código responsable de soportar los modelos y controladores de la aplicación, mientras que en el *front-end* se despliega básicamente el código responsable de la vista, integrándose ambas partes utilizando una API REST sobre HTTP.

Primero se explica brevemente la organización de la parte de *back-end*:

El código del *back-end* se dispone de tres directorios diferentes: `config`, `controllers` y `routes` e incluye el fichero denominado explícitamente `index.js`.

- Este último fichero incluye *express* que permite integrar *Node.JS* y *mysql*, así como importar el fichero `routes.js`.
- En el directorio `config/` se encuentra un fichero que se encarga de crear la conexión con la base de datos de *mysql*.
- En el directorio `routes/` se encuentra un fichero que desempeña la función de *Router* implementada por *Express*. Es esencial para unir la parte de *front* con la parte de *back* gracias al uso de rutas. Una ruta es una asociación entre la URL utilizada en el *front* que, dependiendo del método HTTP para realizar la petición (GET, POST, etc.) con una función específica del modelo ubicado en el directorio con los controladores (`controllers/`) del proyecto. Dependiendo de la ruta utilizada en el *front* el sistema lleva a cabo un determinado método de solicitud HTTP (GET, POST, etc.) y un modelo ubicado en el controlador del proyecto.
- Finalmente, en el directorio `controllers/` se encuentra un controlador por cada clase del proyecto, los cuales tienen consigo todos los modelos necesarios para el funcionamiento del sistema.

Así se distribuyen los ficheros:

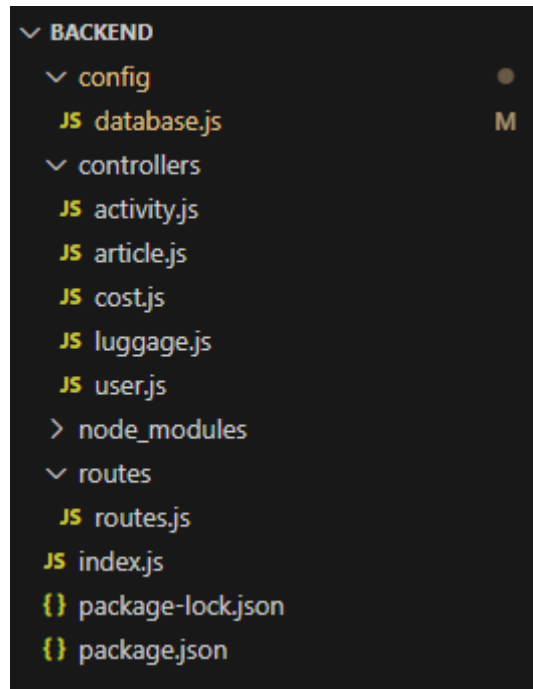


Ilustración 46 - Distribución ficheros back-end

Y, para terminar, se explica la distribución de los ficheros en el *front-end*, que como se ha mencionado previamente, se corresponde con las vistas de la aplicación siguiendo el patrón MVC. En el repositorio del *front-end* se encuentran un número más alto de ficheros respecto a la parte de *back* ya que conlleva muchas más líneas de código. El contenido fundamental se encuentra dentro del directorio `src/`:

- Directorio `assets/`: ficheros `css` para controlar el diseño del proyecto y el tamaño de los componentes dependiendo del tamaño de la pantalla.
- Directorio `components/`: componentes `Vue` para utilizar a lo largo de la aplicación.
- Directorio `public/`: todas las imágenes que se muestran en la página web,
- Directorio `router/`: archivo para implementar la funcionalidad de `vue-router`. Se encarga de controlar las rutas de la url para facilitar la navegabilidad. Cada ruta lleva asociado un nombre y una vista correspondiente al fichero adecuado. También se controla la seguridad en el caso de que no se inicie sesión ya que impide al sistema entrar en la aplicación sin haber introducido un usuario y una contraseña adecuadas.
- Directorio `store/`: un fichero con la funcionalidad de `Vuex` llamada `Store` que es básicamente un contenedor que contiene el estado de su sesión mediante variables globales. Se utiliza principalmente para gestionar la autenticación de los usuarios, usando también el fichero de `vue-router` explicado en el punto anterior.
- Directorio `views/`: ficheros `Vue` dónde se implementan todas las vistas con sus componentes correspondientes.

Así se distribuyen los ficheros:

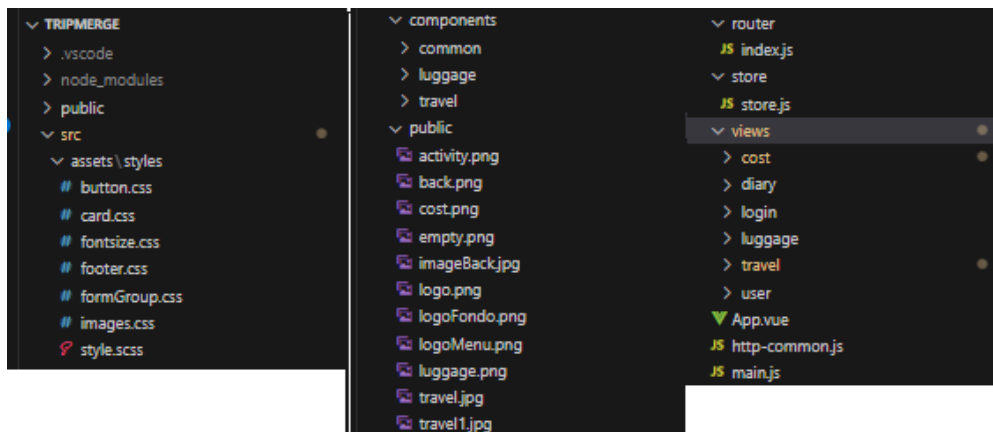


Ilustración 47 - Distribución ficheros front-end

6.2 Autenticación y creación de cuentas de usuario

Para comenzar con la aplicación, es necesario crearse una cuenta de usuario introduciendo todos sus datos: nombre, apellidos, email y contraseña. Cuando el usuario presiona el botón de crear cuenta, el sistema comprueba:

1. Que todos los campos estén completos, ya que son obligatorios.
2. Una vez comprobados, el sistema analiza la contraseña para que tenga un mínimo de 8 caracteres y un máximo de 20.
3. Finalmente, el sistema revisa que no hay un usuario ya creado con el mismo correo, por lo que tiene que ser único.
4. Si todos los pasos anteriores se realizan con éxito, se almacenarán todos los datos en la base de datos. Se ha de comentar que la contraseña se almacena hasheada mediante CRYPTO-JS (Biblioteca para JavaScript de estándares criptográficos) y la función AES para una óptima seguridad y protección de datos.

Tras esos pasos, el usuario tendrá la oportunidad de poder iniciar sesión cuando él quiera y siguiendo casi los mismos pasos que antes. Una vez pulsado el botón de iniciar sesión el sistema comprueba:

1. Que existe un usuario con el email introducido.
2. La contraseña introducida se vuelve a hashear para comprobar que coincide con la contraseña del usuario, previamente hasheada y almacenada en la base de datos.
3. Si los dos valores coinciden, el sistema almacena el email, el nombre y los apellidos del usuario en una `store` que proporciona Vue para llevar a cabo una autenticación en la aplicación.
4. Con esa `store`, el sistema analiza los datos almacenados para que el usuario no pueda introducirse en la aplicación sin haber iniciado sesión, bloquea las rutas de la url para garantizar la seguridad.
5. Finalmente, al cerrar sesión, el sistema eliminará los datos de la `store` para que no vuelva a entrar a la aplicación hasta que se vuelva a iniciar sesión.

6.3 Creación de viajes y sus participantes

Para la utilización de los viajes es interesante explicar la manera en la cual está enfocado este proyecto. Como se puede observar en el diagrama entidad relación (Véase Ilustración 18), la entidad `Actividad` posee una relación recursiva a sí misma puesto que un viaje se considera también una actividad.

En esta aplicación, el usuario podrá añadir actividades dentro de un viaje. Para ello, se enseña a continuación la estructura de la tabla en la base de datos:

id	titulo	descripcion	fecha_inicio	fecha_fin	localizador	id_viaje
3	Bilbao	Viaje a bilbao con mis amigos	2024-04-05	2024-04-07	3-	NULL
4	Cena en hamburguesería	Reserva a las 22:30 a nombre de Adri	2024-04-06	2024-04-06	3-4-	3

Ilustración 48 - Ejemplo creación viajes

Como se puede observar en la ilustración, se puede diferenciar un viaje de una actividad gracias al atributo `id_viaje`. Si ese atributo tiene el valor `NULL` quiere decir que es un viaje principal ya que no apunta a ninguna otra actividad. Si por el contrario tiene algún valor quiere decir que es una actividad que es parte de otra.

Algo parecido encontramos en el atributo `localizador`, ya que informa de todas las actividades a las que pertenece una actividad en cuestión. En el ejemplo anterior, la cena en hamburguesería pertenece al viaje Bilbao puesto que tiene su `id` en `id_viaje` y en `localizador`.

6.4 Inserción de gastos y cálculo de los costes

Otro apartado de la aplicación es la gestión de los gastos. Para ello, se asocia cada gasto a una actividad, no a un viaje. Al añadir un gasto y sus participantes, el sistema obtiene el número total de participantes y divide la cantidad del gasto entre ese número para calcular lo que va gastando cada usuario. La aplicación informa al usuario tanto del gasto total asociado a cada actividad del viaje como lo que se ha gastado él, en particular:

Bilbao

Gasto total: 469 €
Gasto individual: 193 €

Cena en hamburguesería
+ Añadir gasto

Tu gasto total de la actividad: 107

Concepto	Gasto individual	Pagador	Gasto total	Participantes	
Cena	72 €	Rubén Velasco Plaza	215 €	- Dynah Staff - Milka Chestney - Rubén Velasco Plaza	
bebida	4 €	Rubén Velasco Plaza	12 €	- Dynah Staff - Milka Chestney - Rubén Velasco Plaza	
Cocktail	24 €	Rubén Velasco Plaza	48 €	- Dynah Staff - Rubén Velasco Plaza	
Postre	7 €	Rubén Velasco Plaza	20 €	- Dynah Staff - Milka Chestney - Rubén Velasco Plaza	

<
1
>

Ilustración 49 - Ejemplo inserción de gasto

6.4 Gestión de actividades

Finalmente, para la gestión de actividades se ha utilizado la librería `fullcalendar` para poder insertar un calendario de gran formato para que el usuario pueda ser informado de todas las actividades que tiene su viaje. El sistema analiza todas las actividades que están relacionadas con el viaje y se las envía al calendario en forma de eventos, con sus títulos, sus fechas de inicio y fin y sus identificaciones.

El usuario puede ver más información pulsando en la actividad en cuestión y tiene la opción de añadir otra actividad dentro de la misma. Al realizar esta acción, los campos para introducir las fechas se ven acotados a las fechas de la actividad padre.

The screenshot shows a calendar application for Bilbao in April 2024. The interface includes a header with the city name 'Bilbao' and a '+ Añadir actividad' button. The calendar grid shows days from 1 to 12. Activities are represented as colored bars: 'Turismo' (blue) and 'Gastronomía' (red). On Saturday, April 6th, there are two overlapping activities: 'Turismo' and 'Cena en hamburgu' (purple). On Sunday, April 7th, there is a 'Gastronomía' activity. On Monday, April 8th, there are 'Turismo' and 'Gastronomía' activities. On Thursday, April 18th, the day is highlighted in yellow.

lun	mar	mié	jue	vie	sáb	dom
1	2	3	4	5	6 Turismo Cena en hamburgu	7 Gastronomía
8 Turismo Gastronomía	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

Ilustración 50 - Ejemplo gestión de actividades

Capítulo 7: PRUEBAS

En este capítulo se realizan las pruebas pertinentes para comprobar el buen funcionamiento de la aplicación web. Para ello, se divide el capítulo en dos partes significativas: pruebas de caja blanca y pruebas de caja negra.

7.1 Pruebas de caja blanca

Las pruebas de caja blanca, también conocidas como pruebas de caja clara o pruebas estructurales, son una técnica de prueba de software que se centra en examinar el código interno de un programa. En contraste con las pruebas de caja negra (que se centran en el comportamiento externo del software), las pruebas de caja blanca implican el conocimiento detallado del código fuente y la lógica interna de un sistema.

Para ello, se realizan las siguientes pruebas durante el proceso de pruebas en los diferentes *sprints*:

- Validación de la conexión con la base de datos.
- Validación de la entrada y salida de los datos con el servidor de la base de datos.
- Validación del manejo de errores por si existiera un error con la conexión.
- Validación de la autenticación y creación de cuentas de usuario.
- Validación del manejo de las rutas introducidas por el usuario.
- Control de sesiones.

7.2 Pruebas de caja negra

En este segundo apartado del capítulo de pruebas, se llevan a cabo las pruebas de caja negra. Dichas pruebas consisten en comprobar que un sistema, o aplicación en este caso, funcione correctamente sin poseer necesariamente conocimientos en el ámbito de la informática. Los comprobantes del funcionamiento se limitan a introducir entradas de datos a la aplicación y recibir unas salidas u otras dependiendo de los anteriores datos. El objetivo principal de esta función es lograr una interacción entre los usuarios y el sistema.

Para realizar estas pruebas de caja negra es necesario contar con uno o varios responsables que no hayan participado en la creación del proyecto, por lo que, en mi caso, he convocado a varios de mis compañeros de carrera en la Universidad de Valladolid para llevar a cabo esta acción. A continuación, se muestra un listado de tablas con las distintas pruebas:

PCN-01 Crear cuenta de usuario	
Objetivo	Crear una cuenta de usuario con los datos introducidos.
Precondiciones	N/A
Datos de entrada	Nombre: Sergio Apellidos: Calvo de Santos Email: sergio@gmail.com Contraseña: sergiocalvo
Acción esperada	Al pulsar el botón de ‘Crear cuenta’ se mostrará una notificación verde con un mensaje y se iniciará sesión automáticamente.
Resultado	Positivo

Tabla 56 - PCN-01 Crear cuenta de usuario

PCN-02 Iniciar sesión	
Objetivo	Iniciar sesión a la aplicación con el email y la contraseña del usuario.
Precondiciones	Haber creado una cuenta de usuario
Datos de entrada	Email: sergio@gmail.com Contraseña: sergiocalvo
Acción esperada	Al pulsar el botón de ‘Iniciar sesión’ se mostrará una notificación exitosa y se abrirá la aplicación.
Resultado	Positivo

Tabla 57 - PCN-02 Iniciar sesión

PCN-03 Añadir un viaje	
Objetivo	Crear un nuevo viaje
Precondiciones	Haber iniciado sesión.
Datos de entrada	Título: Viaje a Argentina Descripción: Viaje creado para visitar monumentos en Argentina, probar su gastronomía, ver sus calles, hablar con amigos... Se hará a finales de año algo antes de Navidad. Vamos con Rubén, Sergio, Ismael, Fernando... Cogeremos un avión y nos moveremos en coche. Fecha de inicio: 01/11/2024 Fecha de fin: 31/11/2024 Participantes: sergio@gmail.com ruben@gmail.com
Acción esperada	Al pulsar el botón de ‘Añadir viaje’ se mostrará una notificación exitosa y se creará el nuevo viaje con los datos anteriores.
Resultado	Positivo

Tabla 58 - PCN-03 Añadir un viaje

PCN-04 Modificar datos de un viaje	
Objetivo	Cambiar los datos de un viaje
Precondiciones	Haber creado el viaje.
Datos de entrada	Título: Viaje a México Descripción: Viaje creado para visitar monumentos en México, probar los tacos, ver sus calles, hablar con amigos... Se hará a finales de verano algo antes de septiembre. Vamos con Rubén, Sergio, Ismael, Fernando... Cogemos un avión y nos moveremos en bici. Fecha de inicio: 31/07/2024 Fecha de fin: 30/08/2024 Participantes: sergio@gmail.com
Acción esperada	Al pulsar el botón de 'Guardar cambios' se mostrará una notificación exitosa y se modificarán los datos del viaje con los datos anteriores.
Resultado	Positivo

Tabla 59 - PCN-04 Modificar datos de un viaje

PCN-05 Eliminar un viaje	
Objetivo	Eliminar un viaje
Precondiciones	Haber creado el viaje.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón de 'Eliminar viaje' se mostrará una ventana de confirmación y después una notificación exitosa. Finalmente se borrarán todos los datos del viaje y sus predecesores.
Resultado	Positivo

Tabla 60 - PCN-05 Eliminar un viaje

PCN-06 Añadir una actividad	
Objetivo	Crear una nueva actividad asociado a un viaje.
Precondiciones	Haber creado un viaje.
Datos de entrada	Título: Vuelo en avión Descripción: Vuelo desde el aeropuerto de Barajas a las 10:00 en la terminal 4 Fecha de inicio: 31/07/2024 Fecha de fin: 01/08/2024 Participantes: sergio@gmail.com ruben@gmail.com
Acción esperada	Al pulsar el botón de 'Añadir actividad' se mostrará una notificación exitosa y se añadirá la actividad a la base de datos.
Resultado	Positivo

Tabla 61 - PCN-06 Añadir una actividad

PCN-07 Modificar datos de una actividad	
Objetivo	Cambiar los datos de un viaje
Precondiciones	Haber creado la actividad.
Datos de entrada	Título: Viaje en barco Descripción: Viaje en barco a las 12:30 en el puerto 2 Fecha de inicio: 02/08/2024 Fecha de fin: 04/08/2024 Participantes: sergio@gmail.com ruben@gmail.com
Acción esperada	Al pulsar el botón de ‘Guardar cambios’ se mostrará una notificación exitosa y se modificarán los datos de la actividad con los datos anteriores.
Resultado	Positivo

Tabla 62 - PCN-07 Modificar datos de una actividad

PCN-08 Eliminar una actividad	
Objetivo	Eliminar una actividad
Precondiciones	Haber creado la actividad.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón de ‘Eliminar viaje’ se mostrará una ventana de confirmación y después una notificación exitosa. Finalmente se borrarán todos los datos del viaje y sus predecesores.
Resultado	Positivo

Tabla 63 - PCN-08 Eliminar una actividad

PCN-09 Añadir un gasto	
Objetivo	Crear un nuevo gasto a una actividad
Precondiciones	Haber creado la actividad.
Datos de entrada	Concepto: Billeto de barco Cantidad: 120€ Pagador: ruben@gmail.com Participantes: sergio@gmail.com ruben@gmail.com
Acción esperada	Al pulsar el botón de ‘Añadir gasto’ se mostrará una notificación exitosa y se añadirá el gasto a la base de datos.
Resultado	Positivo

Tabla 64 - PCN-09 Añadir un gasto

PCN-10 Modificar un gasto	
Objetivo	Modificar un gasto de una actividad
Precondiciones	Haber creado el gasto.
Datos de entrada	Concepto: Billetes con descuento Cantidad: 50€ Pagador: sergio@gmail.com Participantes: sergio@gmail.com ruben@gmail.com
Acción esperada	Al pulsar el botón de 'Guardar cambios' se mostrará una notificación exitosa y se añadirá el gasto a la base de datos.
Resultado	Positivo

Tabla 65 - PCN-10 Modificar un gasto

PCN-11 Eliminar un gasto	
Objetivo	Eliminar un gasto de una actividad
Precondiciones	Haber creado el gasto.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón de 'Eliminar gasto' se mostrará una ventana de confirmación y después una notificación exitosa. Finalmente se borrarán todos los datos del viaje y sus predecesores.
Resultado	Positivo

Tabla 66 - PCN-11 Eliminar un gasto

PCN-12 Añadir artículos en el equipaje	
Objetivo	Añadir artículos en el equipaje de un viaje
Precondiciones	Haber creado un viaje.
Datos de entrada	Artículos adquiridos: <ul style="list-style-type: none"> • Camiseta • Gel de ducha • Sudadera • Cargador • Móvil • DNI Artículos no adquiridos: <ul style="list-style-type: none"> • Pasaporte • Calcetines • Pantalones • Colonia
Acción esperada	Al pulsar el botón de 'Guardar cambios' se mostrará una notificación exitosa y se añadirá la actividad a la base de datos.
Resultado	Positivo

Tabla 67 - PCN-12 Añadir artículos en el equipaje

PCN-13 Eliminar un artículo	
Objetivo	Eliminar un artículo del equipaje de un viaje
Precondiciones	Haber creado el viaje.
Datos de entrada	N/A
Acción esperada	Al pulsar el icono de la papelera en el artículo a eliminar, se mostrará una ventana de confirmación y después una notificación exitosa. Finalmente se borrará el artículo en cuestión.
Resultado	Positivo

Tabla 68 - PCN-13 Eliminar un artículo

PCN-14 Modificar datos de un usuario	
Objetivo	Modificar los datos de un usuario ya registrado en la aplicación
Precondiciones	Haber creado una cuenta de usuario.
Datos de entrada	Nombre: Ismael Apellidos: García Hammad Email: isma@gmail.com Contraseña: 87654321
Acción esperada	Al pulsar el botón de 'Guardar cambios' se mostrará una notificación exitosa y se añadirá la actividad a la base de datos.
Resultado	Positivo

Tabla 69 - PCN-14 Modificar datos de un usuario

PCN-15 Eliminar un usuario	
Objetivo	Eliminar un usuario
Precondiciones	Haber creado una cuenta de usuario.
Datos de entrada	N/A
Acción esperada	Al pulsar el botón de 'Eliminar usuario' se mostrará una ventana de confirmación y después una notificación exitosa. Finalmente se borrará el usuario y todos sus datos correspondientes a su email.
Resultado	Positivo

Tabla 70 - PCN-15 Eliminar un usuario

Capítulo 8: DOCUMENTACIÓN DEL PRODUCTO

8.1 Manual de instalación

8.1.1 Instalación y despliegue de aplicación

Para instalar la aplicación *Tripmerge* es necesario seguir todos los siguientes pasos de una manera correcta, sin saltarse ninguno de ellos. Para facilitar su instalación se proporcionan sendos proyectos de GitHub con todo el código necesario.

1. **Instalación Software que constituye la plataforma de ejecución de la aplicación:** Es necesario obtener el software instalándolo siempre desde sus páginas oficiales. Los programas utilizados son Node.JS y MYSQL Server.
2. **Despliegue del software de la aplicación *TripMerge*:** A partir de los dos repositorios del código proporcionados en GitHub, el usuario debe clonar esos repositorios en su dispositivo. Para ello, ha de crearse una carpeta en el disco duro C llamada `tripmerge/` y acceder a ella. Se proporciona a continuación los comandos pertinentes para realizar esas acciones:

```
$ mkdir C:\tripmerge
$ cd C:\tripmerge
```

Luego, se introducen los siguientes comandos para obtener los dos directorios de todo el código de la aplicación:

```
$ git clone https://github.com/rubenv99/tripmerge.git
$ git clone https://github.com/rubenv99/tripmergeBack.git
```

3. Una vez obtenido los dos repositorios, se deben instalar todas las dependencias y librerías de cada uno de ellos introduciendo los siguientes comandos en sus correspondientes carpetas:

```
$ cd C:\tripmerge\tripmerge
$ npm install

$ cd C:\tripmerge\tripmergeBack
$ npm install
```

4. Lanzar MYSQL Command Line Client, configurarlo con la contraseña 'tripmerge' y el puerto 3306.
5. Iniciar MYSQL Command Line Client y crear la base de datos mediante el siguiente comando:

```
$ CREATE DATABASE tripmerge;
```

6. Finalmente, se crean todas las tablas importando el fichero `C\tripmergeBack\createTables.sql` con el siguiente comando:

```
$ mysql -u root -p tripmerge < C\tripmergeBack\createTables.sql
```

8.1.2 Ejecución de la aplicación *TripMerge*

Finalmente, tras haber concluido los anteriores pasos, el usuario ya puede arrancar la aplicación. Para ello, se describen brevemente los pasos a seguir:

1. Iniciar `MYSQL Command Line Client` en su terminal mediante el siguiente comando e introducir la contraseña 'tripmerge':

```
$ mysql -u root -p
```

2. Ejecutar el fichero `start-backend.bat` ubicado en el repositorio `tripmergeBack` y el fichero `start-frontend.bat` en el directorio `tripmerge`.
3. Para concluir, el usuario debe introducir la URL `http://localhost:8080/` en el navegador para ejecutar la aplicación.

8.2 Manual de usuario

Para comenzar a utilizar alguna aplicación, es necesario documentarse antes para conocer el funcionamiento de ella. Por lo tanto, en este apartado se procede a explicar el manual para el usuario. Las instrucciones se dividirán en los cinco apartados significativos de este proyecto:

1. Creación de cuenta e inicio de sesión
2. Gestión de viajes
3. Gestión de gastos
4. Gestión de equipaje
5. Gestión de agenda de actividades

8.2.1 Creación de cuenta e inicio de sesión

Primero de todo, el usuario deberá seguir los pasos explicados en el apartado anterior 8.1 instalando y arrancando el proyecto.

A continuación, le aparecerá la primera vista de la página para iniciar sesión o, si no dispone de una cuenta, creársela.

1. Para crearse una cuenta de usuario, es necesario rellenar todos los campos del formulario.
2. En el campo del email, es obligatorio introducir el '@' y la contraseña debe tener entre 8 y 20 caracteres.
3. Una vez introducidos los datos, se pulsa el botón de 'Crear cuenta'.

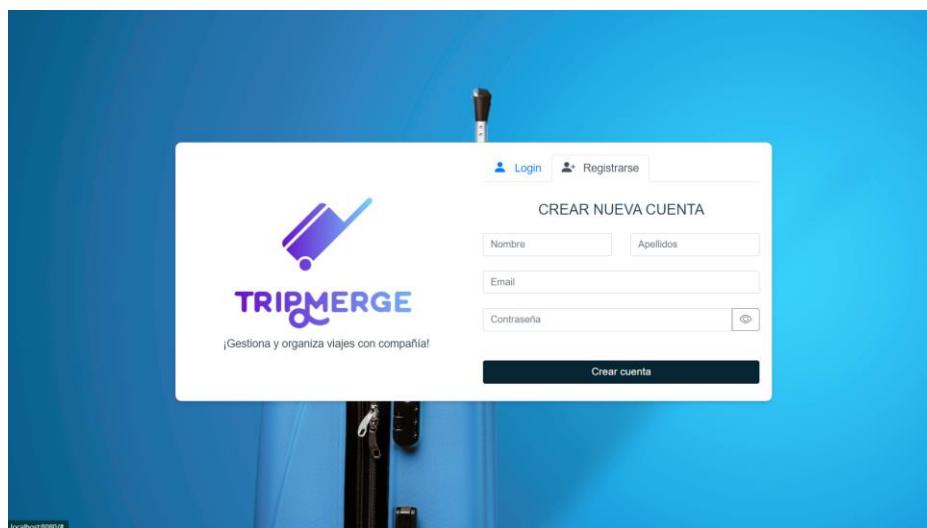


Ilustración 51 - MU - Creación de cuenta de usuario

Tras haber creado la cuenta de usuario, la aplicación inicia sesión automáticamente, pero se procede a explicar la acción de iniciar sesión para cuando el usuario quiera entrar de nuevo.

1. Para iniciar sesión se tiene que pulsar el enlace de 'Login'.
2. A continuación, se rellenan los campos de email y contraseña.
3. Para poder conocer la contraseña introducida se ofrece un botón para hacerla visible.
4. Pulsa el botón de 'Iniciar sesión'
5. Si los datos no corresponden con ningún usuario, se mostrará una notificación y se dará otra oportunidad para poder iniciar sesión.
6. Si los datos son correctos, se introducirá con éxito a la aplicación.

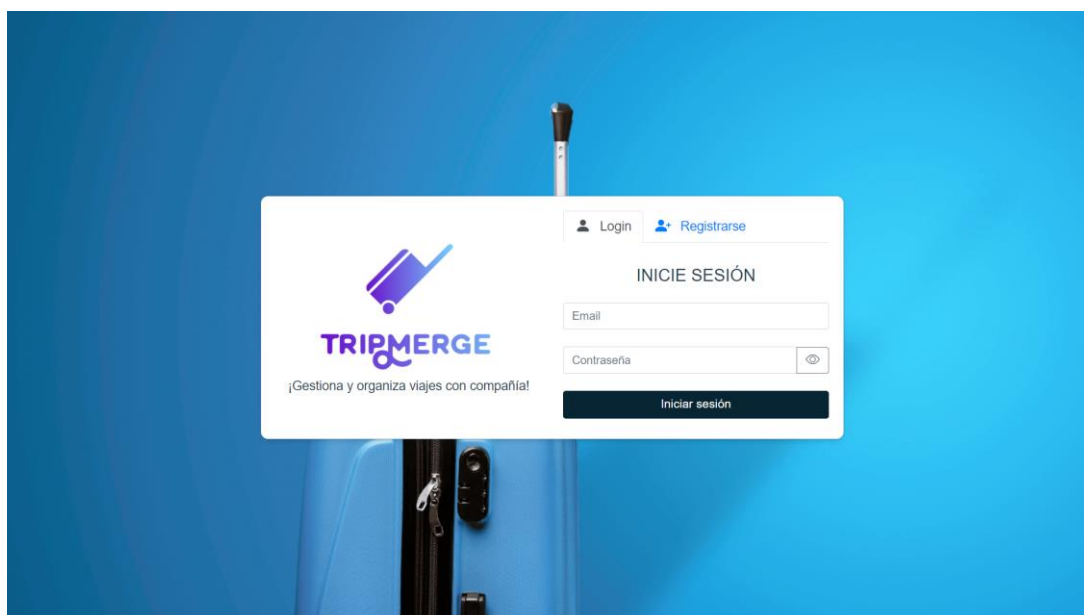


Ilustración 52 - MU - Inicio de sesión

8.2.2 Gestión de viajes

Una vez iniciado sesión, se visualizará la primera vista que consiste en un listado de todos los viajes principales en los que participa el usuario, ordenados por fechas de más reciente a más antigua. Si no posee todavía ningún viaje, se mostrará la siguiente información en todos los apartados de la página:



Ilustración 53 - MU - Pantalla principal vacía

Para poder utilizar esta aplicación, es necesario añadir un viaje a su listado, por lo tanto, al hacer *clic* en el botón de ‘Añadir viaje’, aparecerá un formulario para introducir los datos del viaje en cuestión:

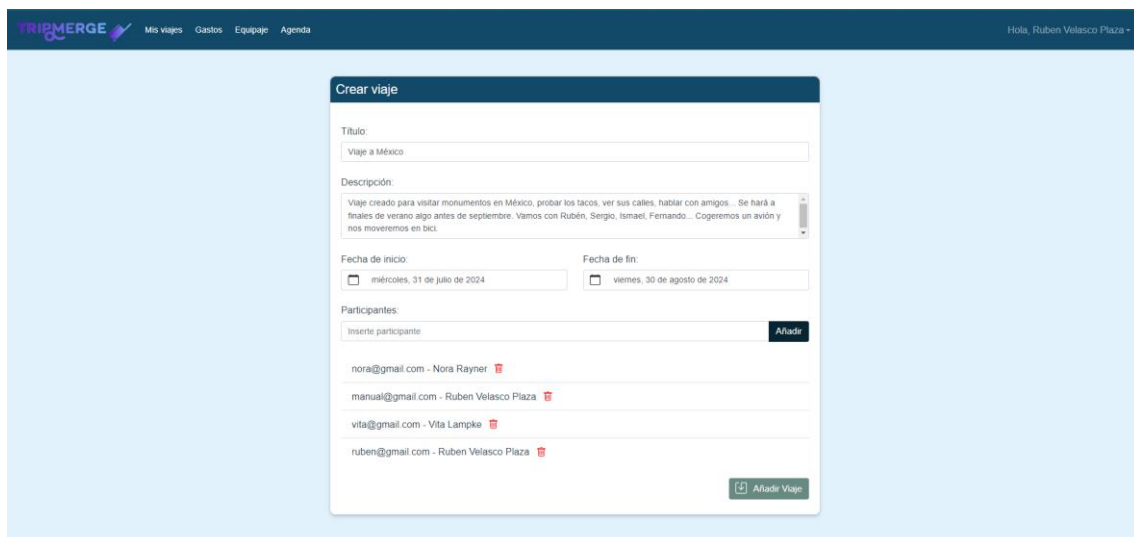


Ilustración 54 - MU - Creación de un viaje

Tras haber rellenado todos los campos, ya que son obligatorios, se pulsa el botón de ‘Añadir viaje’ y se dirigirá a la pantalla principal, donde se podrán observar todos los viajes.

En cada viaje se mostrará el título, la descripción y las fechas, junto a unos accesos directos a otros apartados de la aplicación que se explicarán posteriormente:

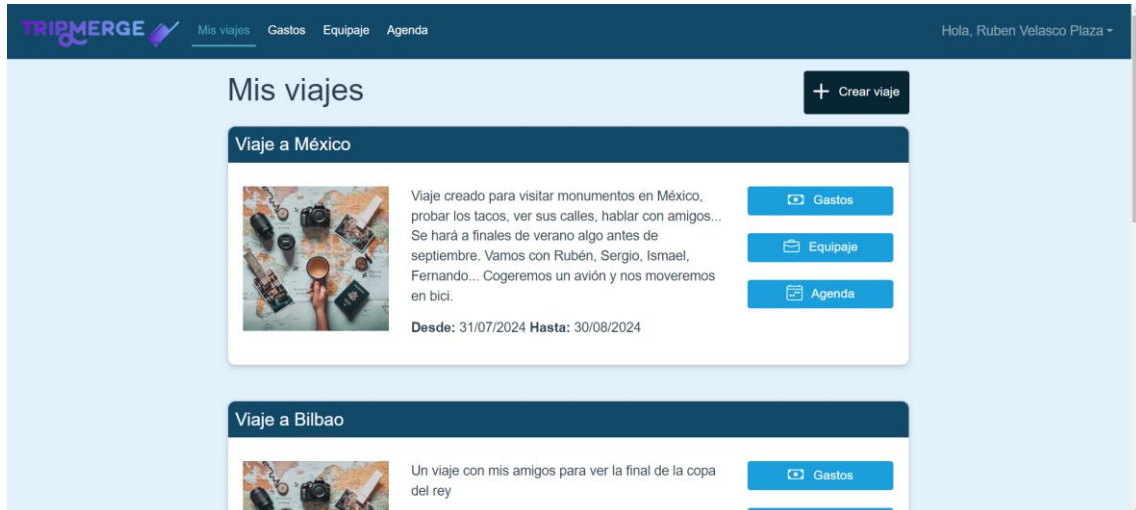


Ilustración 55 - MU - Pantalla principal con datos

Para poder observar los datos de un viaje de una manera más cómoda y amplia, solo es necesario pulsar el viaje en cuestión y se mostrará el siguiente apartado. En dicha ventana, aparecerán dos botones para modificar los datos del viaje y para eliminarlo completamente:

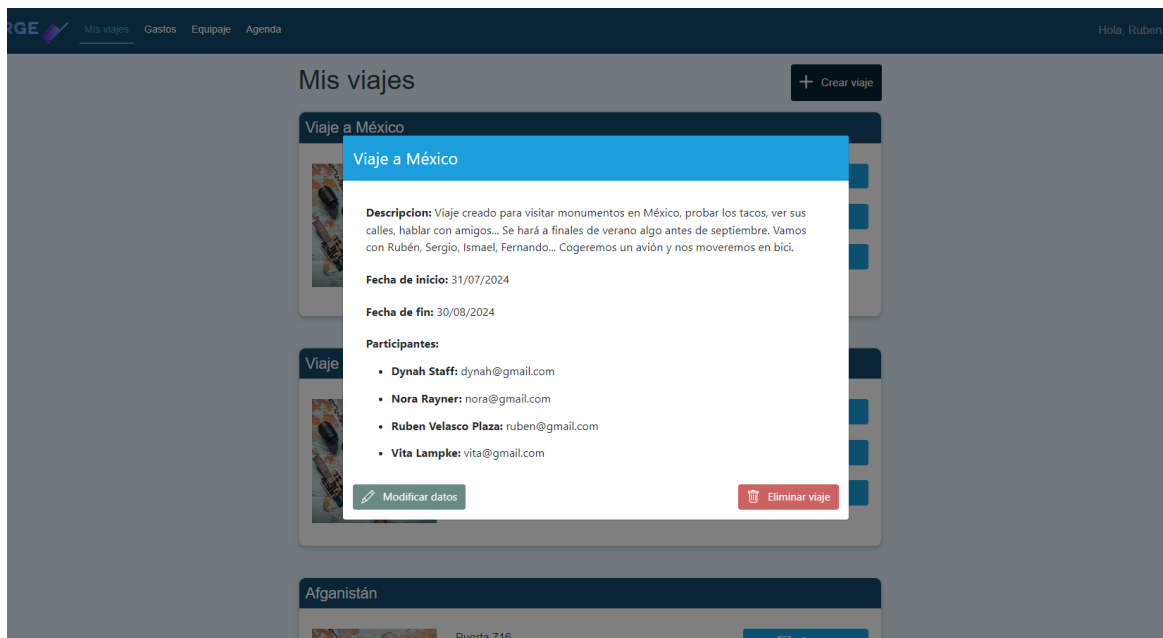


Ilustración 56 - MU - Datos de un viaje

Al pulsar el botón de ‘Eliminar viaje’, saldrá una pequeña ventana de confirmación para evitar errores del usuario:

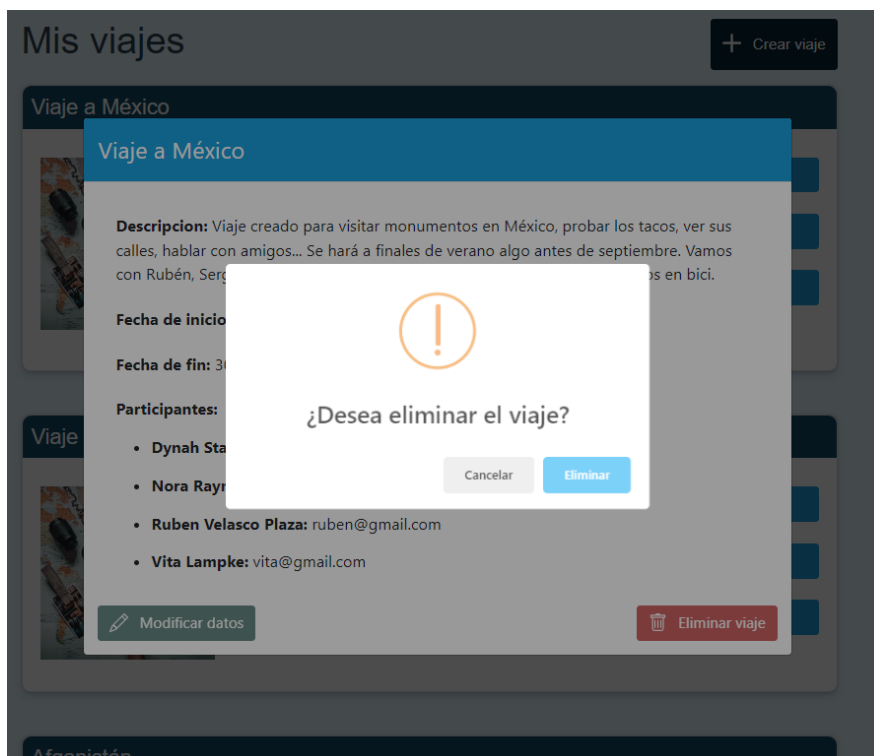


Ilustración 57 - MU - Eliminación de un viaje

Y, para terminar con la parte de los viajes principales, al pulsar el botón de ‘Modificar datos’, se redirigirá a una vista parecida a una de las anteriores, en la cual se podrán introducir los nuevos datos del viaje, como cambiar sus participantes:

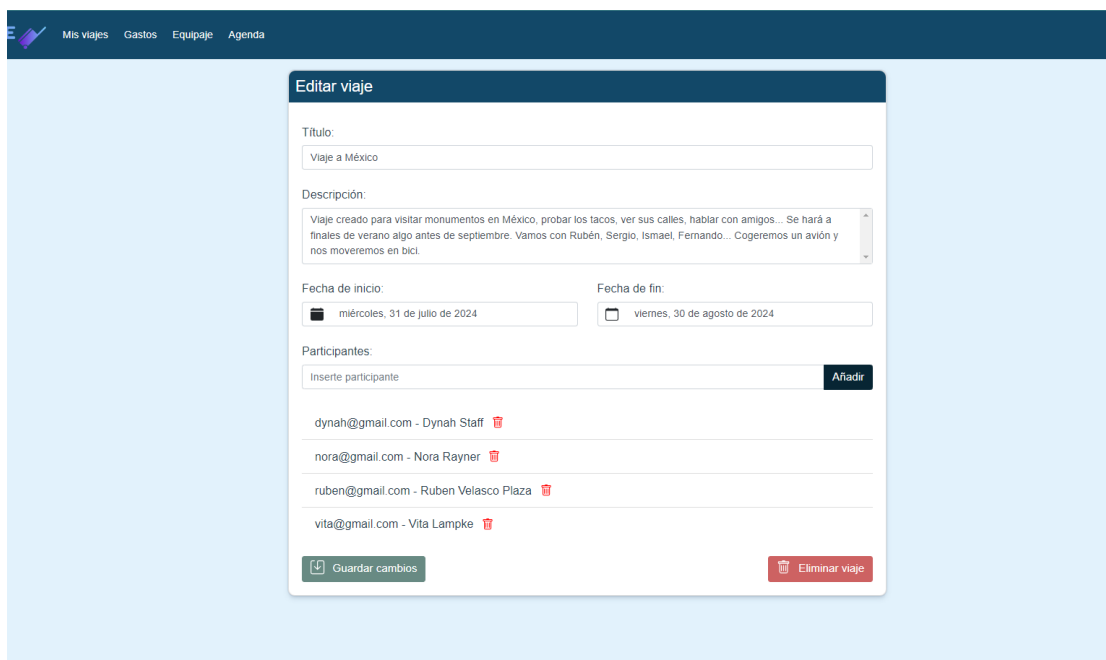


Ilustración 58 - MU - Editar viaje

8.2.3 Gestión de gastos

A continuación, se explicará la parte de Gastos y, para acceder a ella, se utilizará el menú de navegación horizontal que se encuentra en la parte superior de la pantalla. Se mostrará una tabla con todos los viajes del usuario, con su título, fecha de inicio, fecha de fin y un botón para poder observar todos los gastos asociados al viaje:

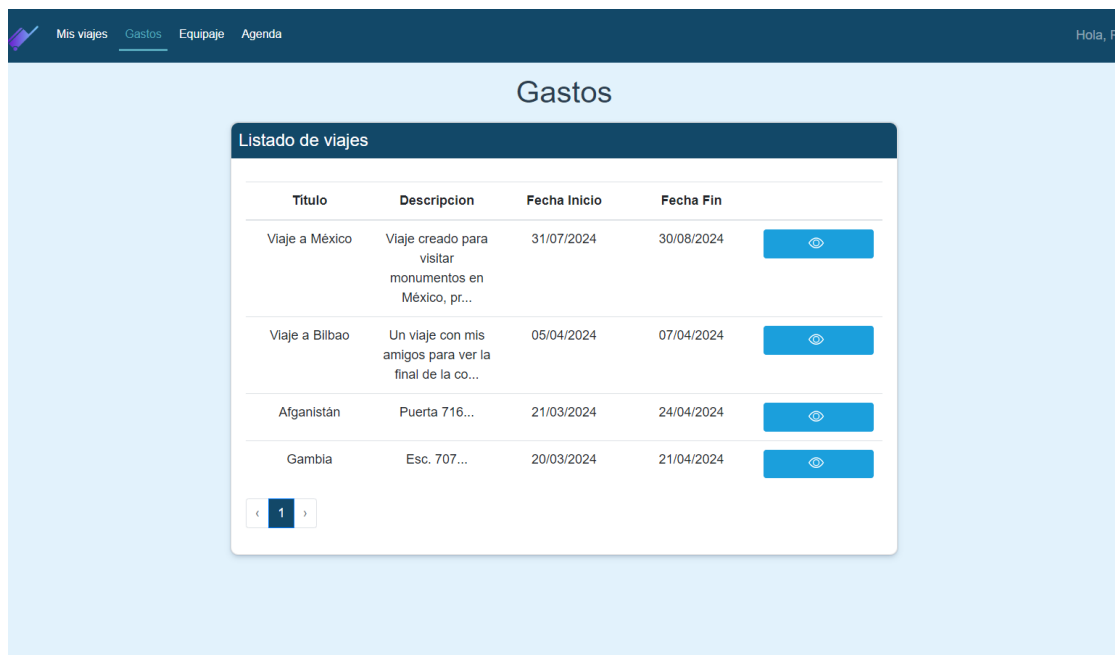


Ilustración 59 - MU - Vista de gastos

Para poder conocer todos los gastos asociados a un viaje el usuario debe hacer *clic* en el botón con el icono de un ojo en la última columna de la tabla:

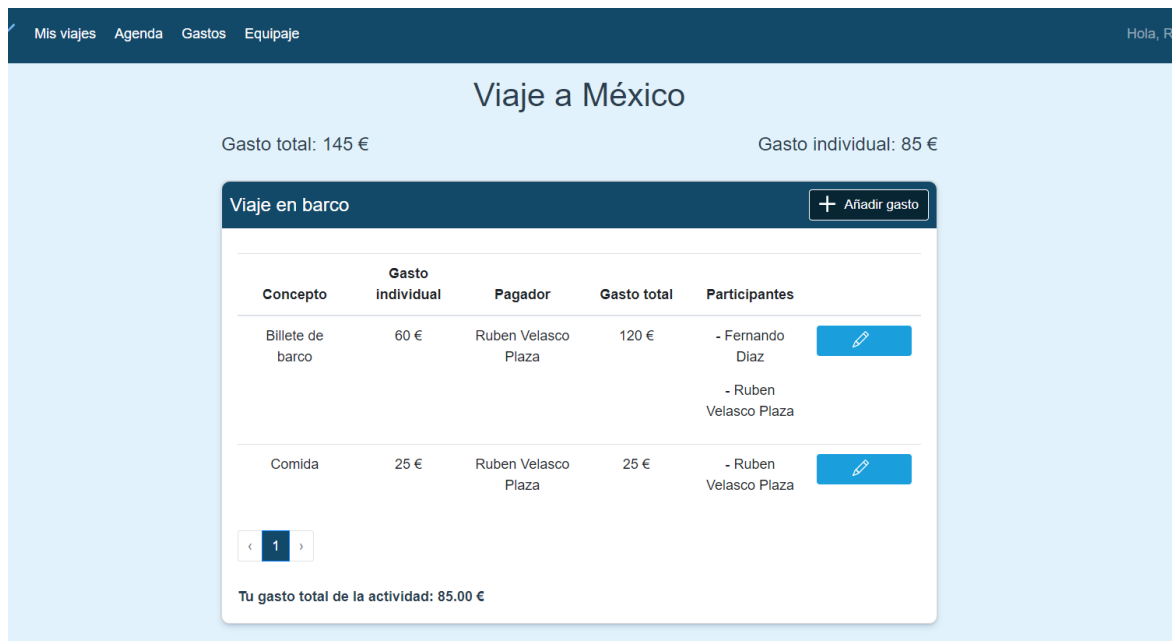


Ilustración 60 - MU - Gastos de un viaje

En esta página, se mostrarán todos los gastos del viaje el cual se ha seleccionado en la vista anterior. El usuario podrá consultar el concepto de los gastos, junto con la cantidad

total del gasto, el pagador y la cantidad que le pertenece al usuario. Al lado se informará de los participantes asociados al gasto y, al lado, un botón para poder editar los datos del gasto:

The screenshot shows a web interface for editing a expense. The top navigation bar includes 'Mis viajes', 'Agenda', 'Gastos', and 'Equipaje'. The main content area is titled 'Editar gasto' and contains the following form elements:

- Concepto:** A text input field containing 'Billete de barco'.
- Cantidad (€):** A text input field containing '120'.
- Pagador:** A dropdown menu with 'Ruben Velasco Plaza' selected.
- Participantes:** A list of checkboxes. 'Fernando Diaz' is unchecked, and 'Ruben Velasco Plaza' is checked.
- Buttons:** A green 'Guardar cambios' button and a red 'Eliminar gasto' button.

Ilustración 61 - MU - Editar gasto

En este apartado, el usuario podrá editar el concepto del gasto, la cantidad total, el pagador y seleccionar los participantes que participan en el gasto en un listado de los usuarios introducidos en el viaje o actividad.

8.2.4 Gestión de Equipaje

La siguiente parte que proporciona la aplicación pertenece a una gestión del equipaje el cual se lleva el usuario a cada viaje.

The screenshot shows the 'Equipaje' section of the application. The top navigation bar includes 'Mis viajes', 'Gastos', 'Equipaje', and 'Agenda'. The main content area is titled 'Equipaje' and contains a 'Listado de viajes' table with the following data:

Título	Descripción	Fecha Inicio	Fecha Fin	
Viaje a México	Viaje creado para visitar monumentos en México, pr...	31/07/2024	30/08/2024	
Viaje a Bilbao	Un viaje con mis amigos para ver la final de la co...	05/04/2024	07/04/2024	
Afganistán	Puerta 716...	21/03/2024	24/04/2024	
Gambia	Esc. 707...	20/03/2024	21/04/2024	

At the bottom of the table, there is a pagination control showing '1' of 1 items.

Ilustración 62 - MU - Vista de equipajes

Al pulsar el botón con el icono de un ojo aparecerá una ventana emergente la cual proporcionará al usuario toda la información de los artículos que corresponden con el equipaje de ese viaje:

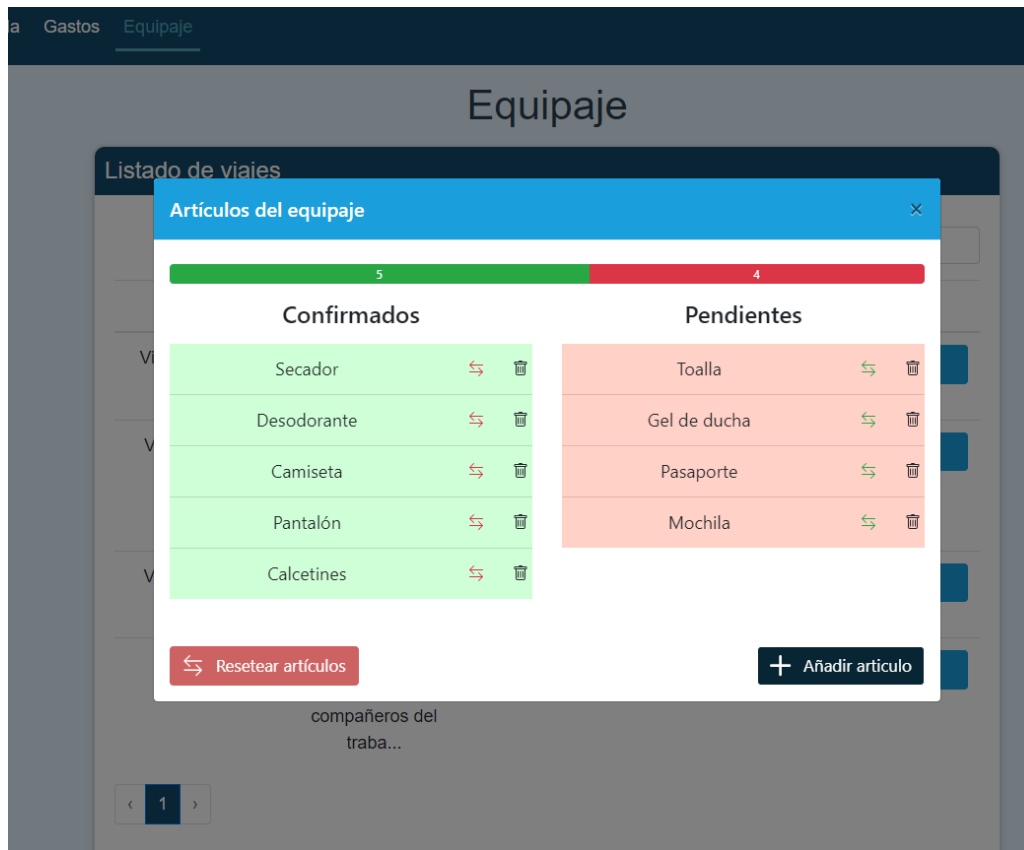
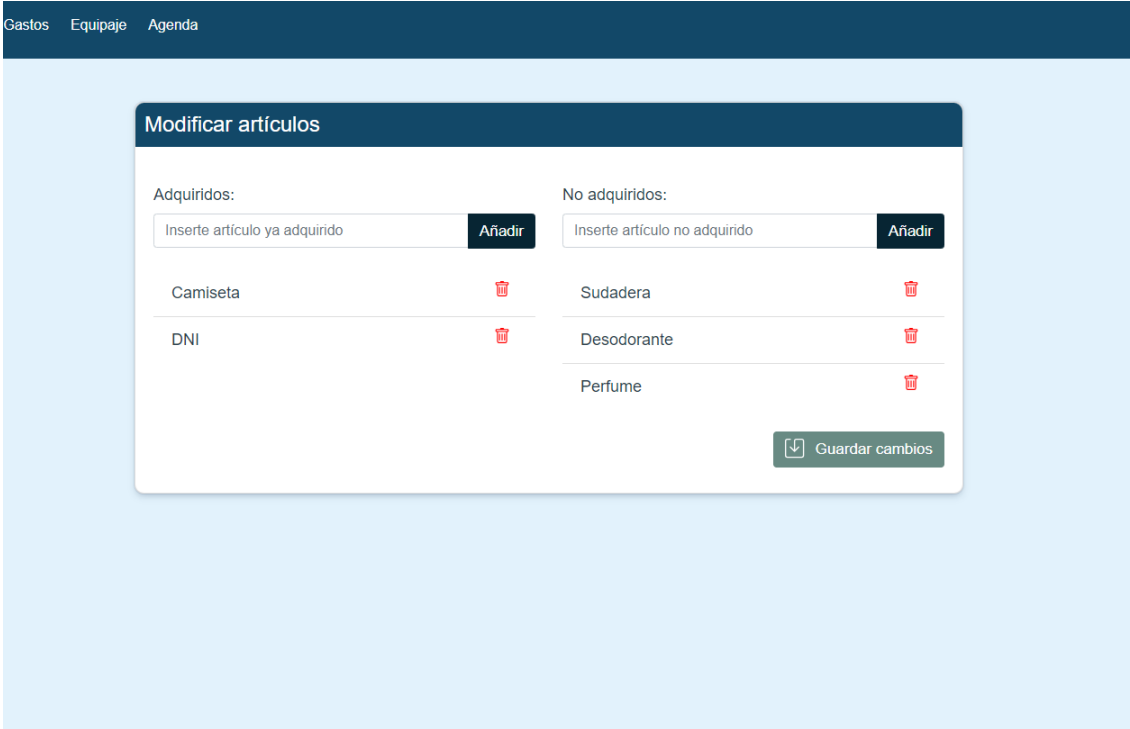


Ilustración 63 - MU - Artículos de un equipaje

En dicha ventana, se dividen todos los artículos en dos partes. La primera columna corresponde a los artículos que el usuario ya ha depositado en la maleta en la vida real y la segunda columna los artículos que le faltan por guardar o por adquirir (ya sea comprándolos o cedidos por otra persona). Para editar de una manera rápida y eficaz se proporcionan dos iconos. El primero, unas flechas, intercambia el artículo de una columna a otra y, el segundo, una papelera, elimina el artículo del equipaje.

Al terminar un viaje, se prepara la maleta para la vuelta a casa. Es por eso que se incluye un botón para resetear todos los artículos y colocarlos en la segunda columna para volver a proceder al recuento de artículos.

Para añadir un artículo nuevo al equipaje es necesario hacer *clic* al botón ‘Añadir artículo’ posicionado abajo a la derecha:



The screenshot shows a web interface with a dark blue header containing the navigation menu: 'Gastos', 'Equipaje', and 'Agenda'. Below the header is a light blue background with a white modal window titled 'Modificar artículos'. The modal is split into two columns. The left column is labeled 'Adquiridos' and contains a text input field with the placeholder 'Inserte artículo ya adquirido' and a dark blue 'Añadir' button. Below this, there are two rows of items: 'Camiseta' and 'DNI', each with a red trash icon to its right. The right column is labeled 'No adquiridos' and contains a text input field with the placeholder 'Inserte artículo no adquirido' and a dark blue 'Añadir' button. Below this, there are three rows of items: 'Sudadera', 'Desodorante', and 'Perfume', each with a red trash icon to its right. At the bottom right of the modal is a dark green button with a checkmark icon and the text 'Guardar cambios'.

Ilustración 64 - MU - Añadir artículos

El usuario introducirá todos los artículos gracias a los dos campos de texto. El primero será para los artículos adquiridos (primera columna en la vista anterior) y el segundo para los no adquiridos (segunda columna). Para guardar dichos artículos, se pulsará el botón de ‘Guardar cambios’ el cual los almacenará en la base de datos y redirigirá la aplicación a la vista anterior.

8.2.5 Gestión de agenda de actividades

Finalmente, la aplicación proporciona un calendario para gestionar las actividades de un viaje. Para acceder a él se pulsará el enlace de ‘Agenda’ del menú de navegación. Ahí se mostrará un listado de todos los viajes:

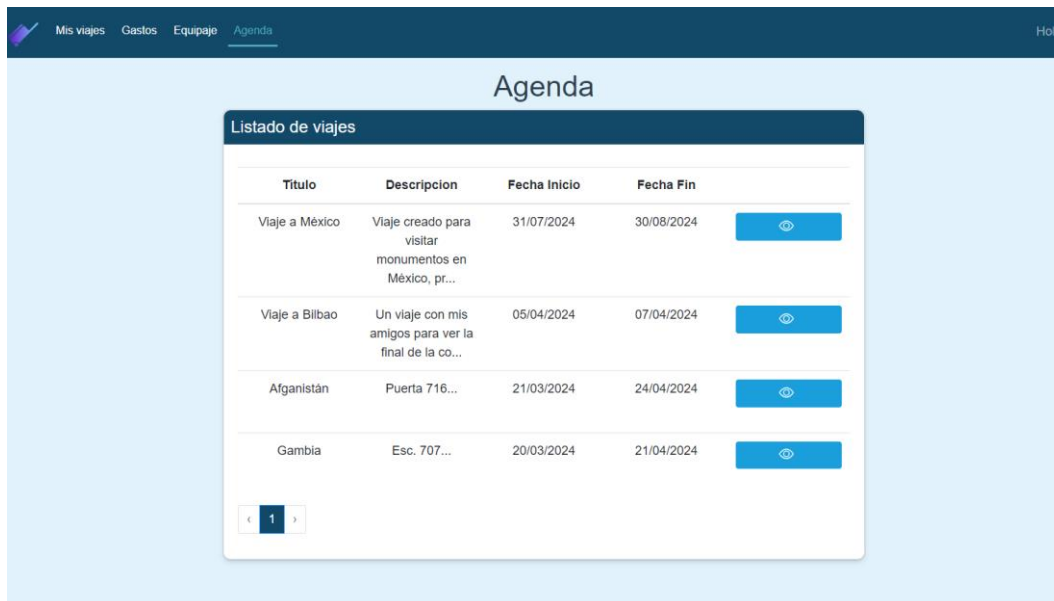


Ilustración 65 - MU – Vista de agenda

Para ver las subactividades de un viaje se hará *clic* en el botón azul, donde aparecerá un calendario con las fechas de las actividades:

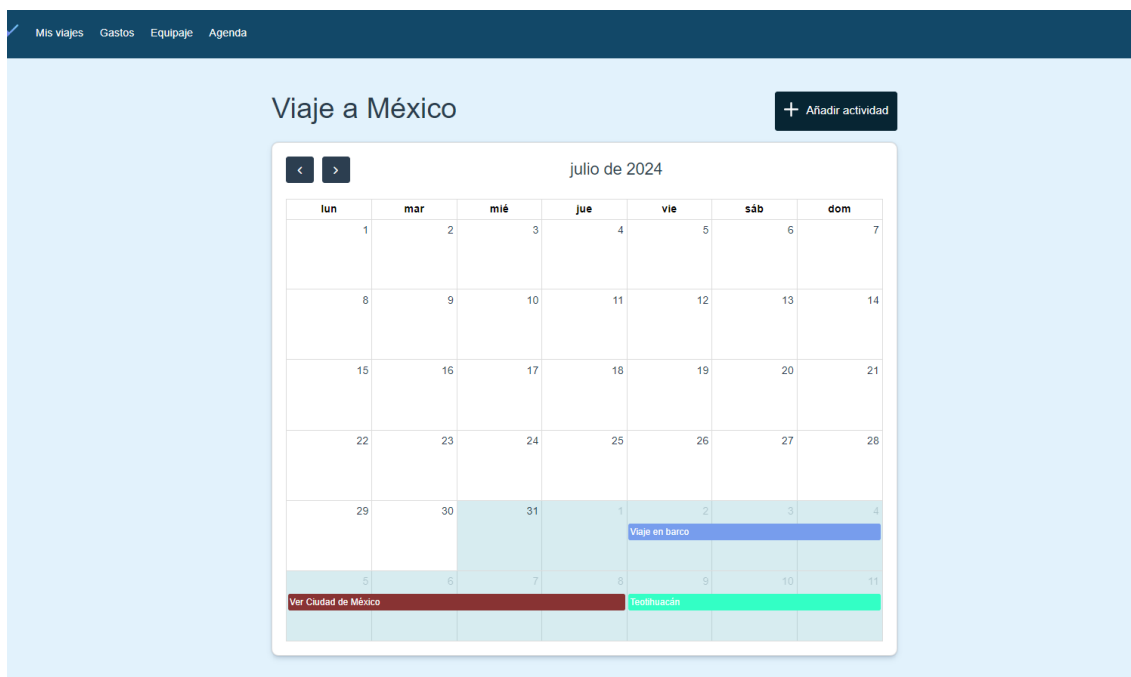
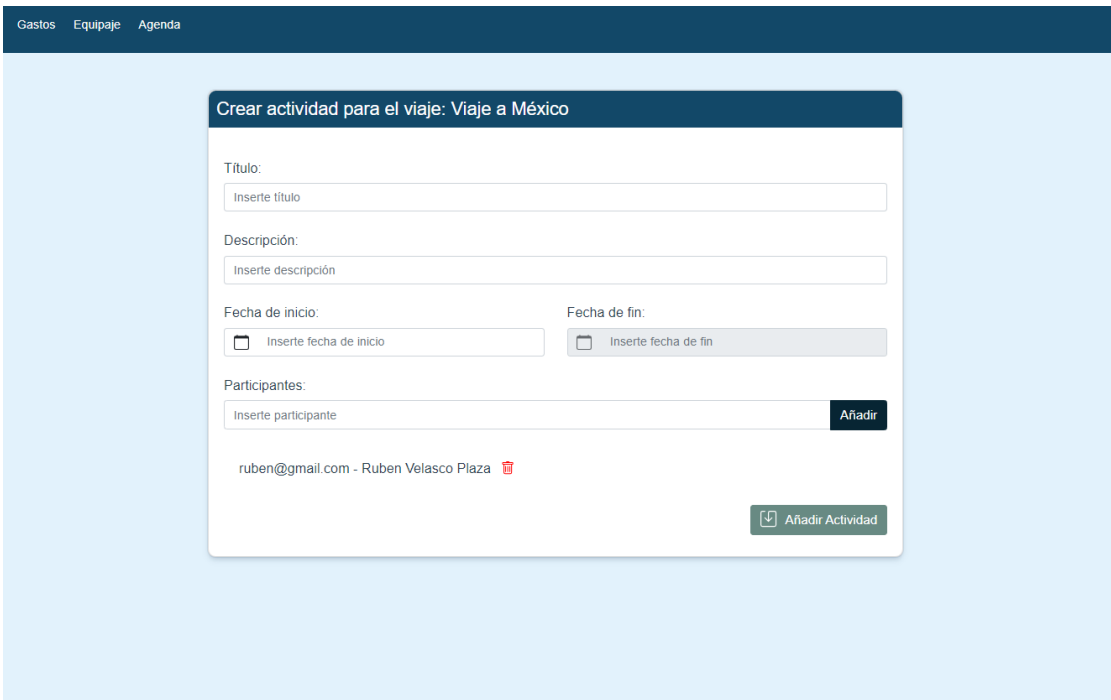


Ilustración 66 - MU - Calendario de actividades

Para poder añadir una nueva actividad, se podrá hacer de dos maneras distintas.

- Si la actividad corresponde con el viaje principal, se hará *clic* en el botón ‘Añadir actividad’ situado en la parte superior a la derecha, ahí se rellenarán todos los campos y se pulsará el botón de ‘Añadir actividad’:

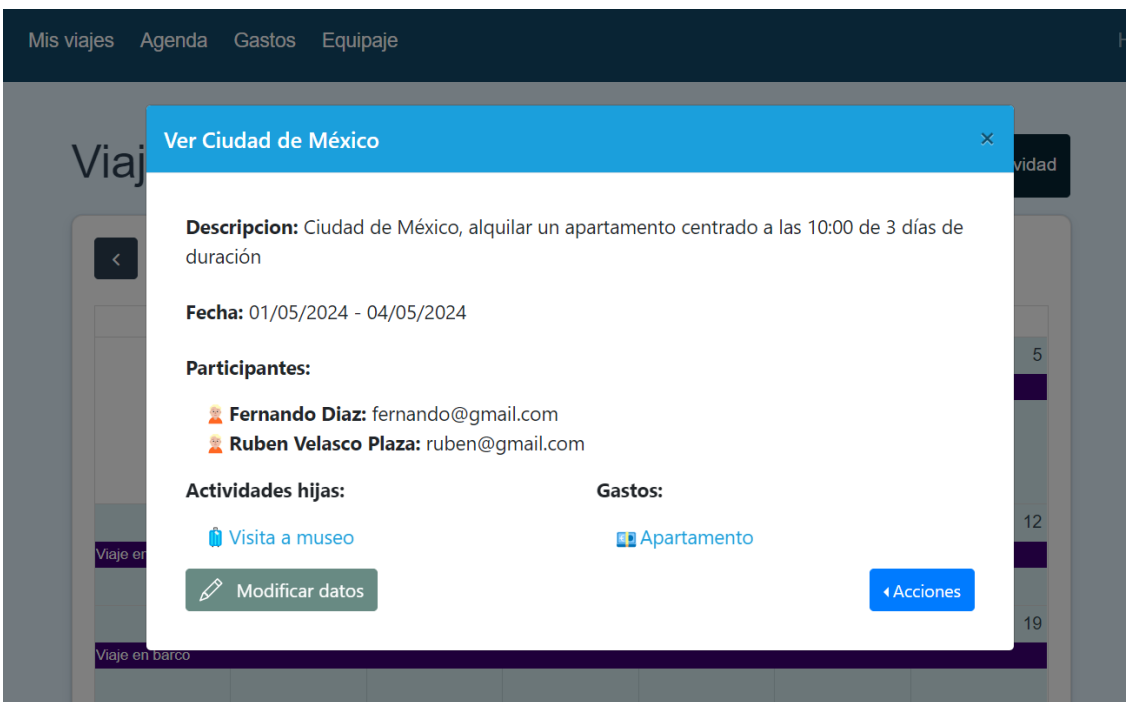


The screenshot shows a web interface for creating an activity. At the top, there are navigation tabs: 'Gastos', 'Equipaje', and 'Agenda'. The main heading is 'Crear actividad para el viaje: Viaje a México'. The form contains the following fields and buttons:

- Título:** A text input field with the placeholder 'Inserte título'.
- Descripción:** A text input field with the placeholder 'Inserte descripción'.
- Fecha de inicio:** A date picker field with the placeholder 'Inserte fecha de inicio'.
- Fecha de fin:** A date picker field with the placeholder 'Inserte fecha de fin'.
- Participantes:** A text input field with the placeholder 'Inserte participante' and an 'Añadir' button next to it.
- Below the participants field, there is a list of existing participants: 'ruben@gmail.com - Ruben Velasco Plaza' with a trash icon.
- At the bottom right of the form, there is a large green button labeled 'Añadir Actividad'.

Ilustración 67 - MU - Crear actividad

- Si la actividad, a su vez, corresponde a otra actividad, se debe pulsar dicha actividad en el calendario donde aparecerá una ventana emergente. Dicha ventana mostrará al usuario todos los datos de esa actividad y varios botones:



The screenshot shows a mobile application interface with a navigation bar at the top containing 'Mis viajes', 'Agenda', 'Gastos', and 'Equipaje'. A modal window titled 'Ver Ciudad de México' is open, displaying the following information:

- Descripción:** Ciudad de México, alquilar un apartamento centrado a las 10:00 de 3 días de duración
- Fecha:** 01/05/2024 - 04/05/2024
- Participantes:**
 - Fernando Díaz:** fernando@gmail.com
 - Ruben Velasco Plaza:** ruben@gmail.com
- Actividades hijas:**
 - Visita a museo
- Gastos:**
 - Apartamento
- At the bottom of the modal, there are two buttons: 'Modificar datos' (with a pencil icon) and 'Acciones' (with a left arrow icon).

Ilustración 68 - MU - Datos de una actividad

- Añadir actividad: Para añadir esa subactividad que corresponde a esta actividad se pulsará el primer botón y se introducirán todos los datos:

The screenshot shows a web interface with a dark blue header containing 'Equipaje' and 'Agenda'. Below the header is a light blue background with a white modal window titled 'Crear actividad para el viaje: Ver Ciudad de México'. The form contains the following fields and controls:

- Título:** A text input field with the placeholder 'Inserte título'.
- Descripción:** A text input field with the placeholder 'Inserte descripción'.
- Fecha de inicio:** A date picker field with the placeholder 'Inserte fecha de inicio'.
- Fecha de fin:** A date picker field with the placeholder 'Inserte fecha de fin'.
- Participantes:** A list of participants. It shows 'ruben@gmail.com - Ruben Velasco Plaza' with a red trash icon to its right. There is an 'Añadir' button to the right of the participant list.
- At the bottom right of the modal is a green button with a checkmark icon and the text 'Añadir Actividad'.

Ilustración 69 - MU - Crear actividad

- Modificar datos: Cambiar los datos que el usuario quiera y seleccionar o deseleccionar los usuarios que participan en dicha actividad:

The screenshot shows a web interface with a dark blue header containing 'Gastos', 'Equipaje', and 'Agenda'. Below the header is a light blue background with a white modal window titled 'Editar actividad'. The form contains the following fields and controls:

- Título:** A text input field containing 'Ver Ciudad de México'.
- Descripción:** A text area containing 'Ciudad de Mexico, alquilar un apartamento centrado a las 10:00 de 3 días de duración'.
- Fecha de inicio:** A date picker field containing 'lunes, 5 de agosto de 2024'.
- Fecha de fin:** A date picker field containing 'jueves, 8 de agosto de 2024'.
- Participantes:** A list of participants with checkboxes:
 - Dynah Staff
 - Nora Rayner
 - Ruben Velasco Plaza
 - Vita Lampke
- At the bottom left is a green button with a checkmark icon and the text 'Guardar cambios'.
- At the bottom right is a red button with a trash icon and the text 'Eliminar actividad'.

Ilustración 70 - MU - Editar actividad

8.2.6 Modificar datos del usuario

Como bonus, la aplicación permite modificar los datos del usuario en cualquier momento. Para ello, el usuario debe pulsar el desplegable de la derecha en el menú de navegación y elegir la opción de ‘Editar usuario’:



Ilustración 71 - MU - Editar usuario en menú de navegación

Una vez ahí, el usuario podrá modificar los campos del nombre, apellidos y la contraseña. Seguidamente, al pulsar de ‘Guardar cambios’, esa información se modificará en la base de datos y se redirigirá a la vista principal:

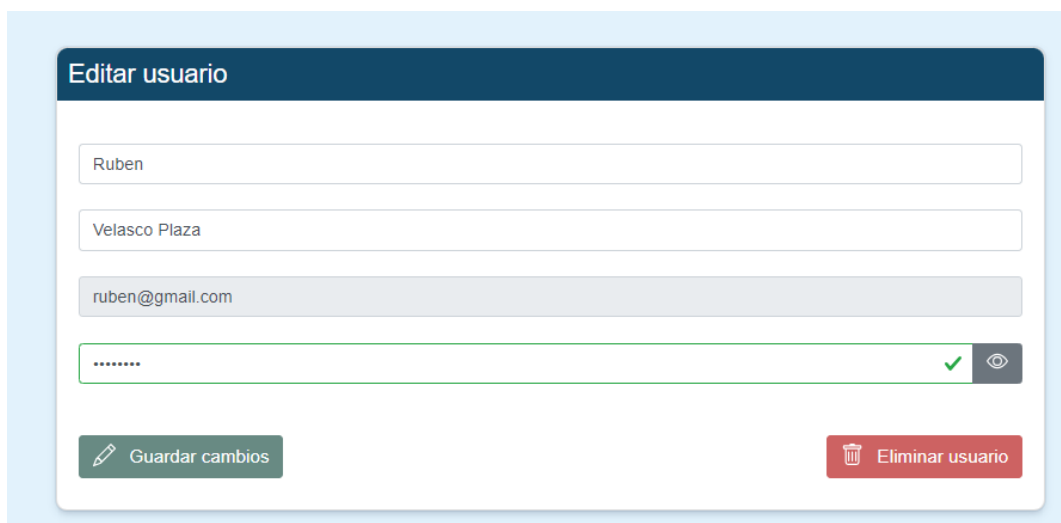


Ilustración 72 - MU - Editar usuario

Y, si el usuario decide borrar su cuenta permanentemente, se pulsará el botón de ‘Eliminar usuario’ donde aparecerá una ventana de confirmación:

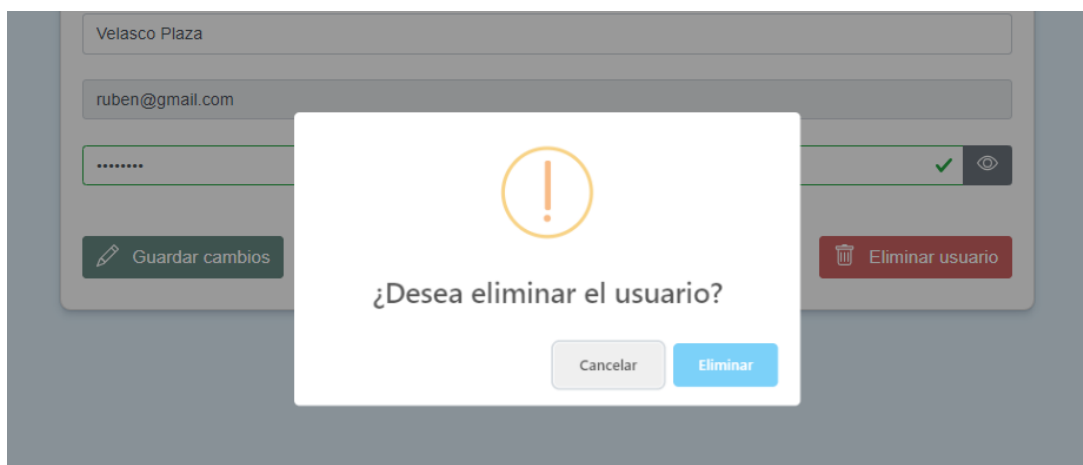


Ilustración 73 - MU - Eliminar usuario

Capítulo 9: CONCLUSIONES Y MEJORAS

El objetivo principal de este proyecto es el de crear una aplicación web sencilla pero útil en el día a día de las personas, centrada en la gestión de los viajes de una persona y su grupo de acompañantes para hacer un seguimiento de cada viaje, centrándose en la gestión de actividades, gastos compartidos y equipaje individual.

El desarrollo de esta se ha centrado en integrar las funcionalidades de una manera intuitiva para que el usuario no pierda demasiado tiempo en intentar entender su funcionamiento. Para ello, se han repartido dichas funcionalidades en cuatro principales apartados, introducidos en el menú de navegación: mis viajes, gastos, equipaje y agenda.

Tras varias pruebas realizadas en la aplicación, considero finalizado este proyecto ya que los objetivos descritos en el Capítulo 1.4 han sido completados, junto a sus criterios de aceptación.

De momento, esta aplicación está desarrollada en local, es decir, que sólo se puede utilizar en mi dispositivo. Pero, al considerar que existen pocas aplicaciones de este estilo y con estas funcionalidades, he pensado en lanzarla al mercado real para analizar su uso y mejorarla poco a poco para un futuro. Tras un buen resultado a mi parecer, tengo la sensación de que va a resultar buena opción para los usuarios que quieran realizar algún viaje con otras personas.

También cabe recalcar, que al ser un trabajo de fin de grado y al tener que ceñirme a un rango de horas, la aplicación web puede ser mejorada insertando más funcionalidades y rediseñar el estilo. Pueden ser varios ejemplos como los siguientes:

- Inserción de más filtros en el listado de todos los viajes para una mejor organización.
- Cálculo de los pagos que deben realizar todos los usuarios al terminar un viaje para conocer cuánto y a quién pagar de una manera ordenada y simplificando los pagos al mínimo.
- Integración de imágenes para el apartado de reservas, por ejemplo, introducir los billetes de avión con el código QR.
- Integración de un apartado para almacenar información de contactos para algún caso de urgencia en un país extranjero.
- Implementar una funcionalidad para que exista la posibilidad de añadir participantes a un viaje mediante un código o un enlace, para facilitar ese proceso.
- Controlar la creación de cuentas verificando el correo electrónico introducido para evitar correos falsos.

Capítulo 10: REFERENCIAS

1. *draw.io - free flowchart maker and diagrams online*. (s. f.). URL: <https://app.diagrams.net/> (Visitado frecuente)
2. *Components | BootstrapVue*. (s. f.). BootstrapVue. URL: <https://bootstrap-vue.org/docs/components> (Visitado frecuente)
3. *GitHub: Let's build from here*. (s. f.). GitHub. URL: <https://github.com/> (Visitado frecuente)
4. *IBM documentation*. (s. f.). URL: <https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/lifecycle-management/6.0.3?topic=requirements-defining-use-cases> (Visitado 25-01-2024)
5. QbD Group. (2022, 21 noviembre). *Requisitos de usuario*. URL: [https://qbdgroup.com/es-es/blog/validacion-de-sistemas-informatizados-requisitos-de-usuario-urs/#:~:text=Los%20Requisitos%20de%20Usuario%20\(RU,%E2%80%9D%2C%20la%20justificaci%C3%B3n%20del%20sistema](https://qbdgroup.com/es-es/blog/validacion-de-sistemas-informatizados-requisitos-de-usuario-urs/#:~:text=Los%20Requisitos%20de%20Usuario%20(RU,%E2%80%9D%2C%20la%20justificaci%C3%B3n%20del%20sistema). (Visitado 29-01-2024)
6. *_ESIC Business & Marketing School*. (s. f.). *Modelo entidad relación: descripción y aplicaciones*. URL: <https://www.esic.edu/rethink/tecnologia/modelo-entidad-relacion-descripcion-aplicaciones> (Visitado 31-01-2024)
7. *Talent.com*. (s. f.). *Búsqueda de empleo en Talent.com | Encuentra vacantes disponibles cerca de ti*. URL: <https://es.talent.com/> (Visitado en 02-02-24)
8. *Método Albrecht para el Análisis de los Puntos Función - manuel.cillero.es*. (2016, 15 octubre). manuel.cillero.es. URL: <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/tecnicas-de-estimacion/metodo-albrecht/> (Visitado en 02-02-24)
9. Machuca, F. (2021, 24 noviembre). *9 software para diseño de interfaces con los que podrás mejorar la experiencia de tus usuarios*. <https://www.crehana.com>. URL: <https://www.crehana.com/blog/transformacion-digital/software-diseno-de-interfaces/> (Visitado en 05-02-2024)
10. Zendesk. (2023, 14 febrero). *¿Qué es la metodología ágil? ¿Para qué sirve?* URL: <https://www.zendesk.com.mx/blog/metodologia-agil-que-es/#:~:text=La%20metodolog%C3%ADa%20%C3%Agil%20es%20un,ha%20el%20t%C3%A9rmino%20del%20proyecto>. (Visitado en 10-02-2024)
11. *Las etapas del ciclo de vida de desarrollo de software según la metodología ágil*. (2018, 10 agosto). URL: <https://www.lucidchart.com/blog/es/ciclo-de-vida-del->

- [desarrollo-de-software-segun-la-metodologia-agil](#) (Visitado 13-02-2024)
12. *MDN Web Docs*. (s. f.). MDN Web Docs. URL: <https://developer.mozilla.org/en-US/> (Visitado 13-02-2024)
 13. *Pinterest*. (s. f.). Pinterest. URL: <https://www.pinterest.es/> (Visitado frecuente)
 14. *BootstrapVue*. (s. f.). GitHub. URL: <https://github.com/bootstrap-vue> (Visitado frecuente)
 15. *RNDGen*. (s. f.). URL: <https://www.rndgen.com/data-generator> (Visitado 25-03-2024)
 16. *DeepL Translate - El mejor traductor del mundo*. (s. f.). URL: <https://www.deepl.com/es/translator> (Visitado frecuente)
 17. *FullCalendar - JavaScript Event Calendar*. (s. f.). URL: <https://fullcalendar.io/> (Visitado 28-03-2024)
 18. Singureanu, C. (2023, 19 abril). Pruebas de caja negra: qué son, tipos, procesos, enfoques, herramientas y mucho más. ZAPTEST. URL: <https://www.zaptest.com/es/pruebas-de-caja-negra-que-son-tipos-procesos-enfoques-herramientas-y-mucho-mas> (Visitado 08-04-2024)
 19. Diagrama de clases - manuel.cillero.es. (2016, 4 diciembre). manuel.cillero.es. URL: <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/> (Visitado en 17-04-2024)
 20. Robledano, A. (2023, 13 abril). Qué es MySQL: Características y ventajas. OpenWebinars.net. URL: <https://openwebinars.net/blog/que-es-mysql/> (Visitado en 26/04/2024)