

# LA RUEDA:

# Optimización de viajes por trabajo

Trabajo de Fin de Máster

Máster Universitario en Ingeniería Informática



VNiVERSiDAD  
DE SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

**SEPTIEMBRE 2021**

**AUTOR:**

Marcos Unzueta Puente

**TUTOR:**

Pablo Chamoso Santos

# Autorización para la presentación del trabajo

D. Pablo Chamoso Santos, personal del Departamento de Informática y Automática de la Universidad de Salamanca.

## Certifica:

Que el trabajo titulado “Optimización de viajes por trabajo” ha sido realizado por D. Marcos Unzueta Puente, con DNI 71036940-Z y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Máster de la titulación Máster en Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a \_\_\_\_\_ de Septiembre de 2019

D. Pablo Chamoso Santos  
Dpto. Informática y Automática  
Universidad de Salamanca

# Índice

Autorización para la presentación del trabajo .....	2
Índice de tablas .....	6
Índice de imágenes.....	7
Resumen.....	9
Palabras clave .....	10
Summary .....	11
Keywords .....	11
1.    Introducción.....	12
2.    Objetivos del proyecto.....	15
2.1.    Objetivos funcionales.....	15
2.1.1.    Gestión de viajeros:.....	15
2.1.2.    Calcular todos los “horarios de rueda” posibles .....	15
2.1.3.    Evaluar todos los horarios de rueda generados.....	15
2.1.4.    Seleccionar el mejor resultado.....	15
2.1.5.    Generar un fichero exportable con el mejor resultado .....	15
2.1.6.    Reiniciar el sistema.....	16
2.2.    Objetivos no funcionales.....	16
2.2.1.    Accesibilidad.....	16
2.2.2.    Funcionamiento en Android.....	16
2.2.3.    Adaptabilidad y escalabilidad.....	17
2.2.4.    Interfaz amigable y sencilla .....	17
2.2.5.    Eficacia .....	17
3.    Estudio de tecnologías similares .....	18
3.1.    Tecnologías similares .....	18
3.1.1.    Dedocar .....	19
3.1.2.    Journify.....	20
3.1.3.    Carpiling .....	25
3.1.4.    BlaBlaCar Daily .....	25
3.2.    Conclusiones .....	26
3.2.1.    Buen nicho de mercado .....	26
3.2.2.    Aplicación única .....	26
3.2.3.    Beneficios .....	27
4.    Técnicas y herramientas.....	28
4.1.    Lenguajes de programación.....	28
4.1.1.    Java.....	28
4.1.2.    SQL .....	29

<i>Marcos Unzueta Puente</i>	<i>LA RUEDA</i>	<i>TFM</i>
4.1.3. XML .....		29
4.2. Herramientas .....		30
4.2.1. Android Studio .....		30
4.2.2. Gradle .....		30
4.2.3. Androidx .....		31
4.2.4. Github .....		32
4.2.5. Android-pdf-viewer .....		32
4.2.6. SQLite .....		33
4.2.7. AVD Manager .....		33
4.2.8. Microsoft Word .....		35
4.2.9. Visual Paradigm .....		35
4.2.10. EZEstimate .....		35
4.2.11. Microsoft Project .....		36
5. Aspectos relevantes del desarrollo .....		37
5.1. Marco de trabajo .....		37
5.2. Planificación temporal .....		38
5.2.1. Fase inicial .....		38
5.2.2. Fase de elaboración .....		40
5.2.3. Fase de construcción .....		43
5.2.4. Fase de transición .....		47
5.3. Especificación de requisitos .....		48
5.3.1. Objetivos .....		48
5.3.2. Requisitos de información .....		49
5.3.3. Requisitos funcionales .....		50
5.3.4. Requisitos no funcionales: .....		52
5.4. Análisis del sistema .....		53
5.4.1. Modelo de dominio .....		53
5.4.2. Vista de arquitectura del modelo de análisis .....		54
5.4.3. Realización de casos de uso del modelo de análisis .....		55
5.5. Diseño del sistema .....		56
5.5.1. Patrón arquitectónico .....		57
5.5.2. Capas del diseño .....		58
5.5.3. Vista arquitectónica .....		59
5.5.4. Modelo de despliegue .....		60
5.5.5. Realización de casos de uso del modelo de diseño .....		60
5.6. Documentación técnica sobre la aplicación .....		62
5.6.1. Manifests .....		62
5.6.2. Java .....		62

<i>Marcos Unzueta Puente</i>	<i>LA RUEDA</i>	<i>TFM</i>
5.6.3.	Res .....	63
5.6.4.	Gradle .....	63
5.7.	Instalación .....	64
5.8.	Funcionalidad de la aplicación .....	64
5.8.1.	Menú principal .....	64
5.8.2.	Viajeros .....	65
5.8.3.	Eliminar viajero .....	66
5.8.4.	Añadir viajero .....	67
5.8.5.	Modificar viajero .....	68
5.8.6.	Generar rueda .....	68
5.8.7.	Compartir rueda .....	70
5.8.8.	Reiniciar rueda .....	71
6.	Conclusiones y líneas de trabajo futuro .....	73
6.1.	Conclusiones .....	73
6.1.1.	Objetivos funcionales .....	73
6.1.2.	Objetivos no funcionales .....	74
6.2.	Aprendizaje .....	74
6.2.1.	Android Studio : .....	75
6.2.2.	Automatización PDF: .....	75
6.2.3.	Cálculos matemáticos: .....	75
6.2.4.	Uso SQLite: .....	75
6.2.5.	Paralelización del trabajo y segundo plano: .....	75
6.2.6.	Documentación: .....	75
6.3.	Trabajo futuro .....	75

# Índice de tablas

Tabla 1: OBJ-003 Gestión de rueda .....	48
Tabla 2: IRQ-004 Información del Horario de Rueda.....	49
Tabla 3: UC-001 Añadir viajero.....	51
Tabla 4: NRF-001 Funcionamiento de la aplicación en el S.O Android.....	52

# Índice de imágenes

Ilustración 1: Media de tiempo empleado para llegar al lugar de trabajo en Europa, estudio de Michael Page .....	12
Ilustración 2: Principales movimientos interprovinciales de Castilla y León .....	13
Ilustración 3: Horario elaborado a mano .....	14
Ilustración 4: Porcentaje de tráfico web por dispositivo en 2020.....	16
Ilustración 5: Sistemas operativos de móvil más utilizados .....	17
Ilustración 6: Gráfica personas que cambian de provincia por trabajo en España .....	18
Ilustración 7:Logo dedocar .....	19
Ilustración 8: Búsqueda compartir vehículo para ir a trabajar .....	19
Ilustración 9: Dedocar en Play Store .....	20
Ilustración 10: Búsqueda compartir vehículo para ir a trabajar .....	20
Ilustración 11: Interfaz journify .....	21
Ilustración 12: Búsqueda Journify App Store .....	22
Ilustración 13: Interfaz inicial de Journify.....	22
Ilustración 14: Interfaz inicial de Journify.....	22
Ilustración 15: Solicitud de teléfono Journify .....	23
Ilustración 16: Error en IOS Journify.....	23
Ilustración 17: Búsqueda Salamanca-Benavente Journify.....	24
Ilustración 18: Búsqueda sin resultados .....	24
Ilustración 19: Búsqueda Toledo-Madrid Journify .....	24
Ilustración 20: Búsqueda Getafe-Madrid Journify.....	25
Ilustración 21: Interfaz Carpling .....	25
Ilustración 22: BlaBlaCar Daily.....	26
Ilustración 23: Reducción de vehículos .....	27
Ilustración 24: Logo Java .....	28
Ilustración 25: Lenguajes de programación más usados .....	28
Ilustración 26: Logo SQL .....	29
Ilustración 27: Porcentaje de uso SQL.....	29
Ilustración 28: Logo XML .....	29
Ilustración 29: Logo Android Studio .....	30
Ilustración 30: Logo Gradle.....	30
Ilustración 31: Funcionamiento Gradle .....	31
Ilustración 32: Logo AndroidX .....	31
Ilustración 33: Explicación AndroidX .....	32
Ilustración 34: Logo GitHub .....	32
Ilustración 35: Logo SQLite .....	33
Ilustración 36: Funcionamiento SQLite .....	33
Ilustración 37: Hardwares simulables del AVD Manager .....	34
Ilustración 38: Teléfono simulado .....	34
Ilustración 39: Logo Microsoft Word.....	35
Ilustración 40: Logo Visual Paradigm.....	35
Ilustración 41: Logo Microsoft Project .....	36
Ilustración 42: Esquema proceso unificado.....	38
Ilustración 43: Inicio, Iteración 1 .....	39
Ilustración 44: Inicio, Iteración 1 Gnatt .....	39
Ilustración 45: Inicio, Iteración 2 .....	39
Ilustración 46: Inicio, Iteración 2 Gnatt .....	40
Ilustración 47: Inicio, Iteración 3 .....	40
Ilustración 48: Inicio, Iteración 3 Gnatt .....	40
Ilustración 49: Elaboración, Iteración 1.....	41

Ilustración 50: Elaboración, Iteración 1 Gnatt .....	41
Ilustración 51: Elaboración, Iteración 2 .....	42
Ilustración 52: Elaboración, Iteración 2 Gnatt .....	42
Ilustración 53: Elaboración, Iteración 3 .....	43
Ilustración 54: Elaboración, Iteración 3 Gnatt .....	43
Ilustración 55: Construcción, Iteración 1 .....	44
Ilustración 56: Construcción, Iteración 1 Gnatt .....	44
Ilustración 57: Construcción, Iteración 2 .....	45
Ilustración 58: Construcción, Iteración 2 Gnatt .....	45
Ilustración 59: Construcción, Iteración 3 .....	46
Ilustración 60: Construcción, Iteración 3 Gnatt .....	46
Ilustración 61: Construcción, Iteración 4 .....	47
Ilustración 62: Construcción, Iteración 4 Gnatt .....	47
Ilustración 63: Transición, Iteración 1 .....	48
Ilustración 64: Transición, Iteración 1 Gnatt .....	48
Ilustración 65: Diagrama de casos de uso de gestión de viajeros .....	50
Ilustración 66: Diagrama de casos de uso de gestión de rueda .....	51
Ilustración 67: Diagrama de casos de uso de gestión de reinicio .....	51
Ilustración 68: Modelo de dominio .....	53
Ilustración 69: Vista arquitectónica .....	54
Ilustración 70: Vista arquitectónica arriba izquierda .....	54
Ilustración 71: Vista arquitectónica abajo izquierda .....	55
Ilustración 72: Vista arquitectónica derecha .....	55
Ilustración 73: Diagrama de secuencia añadir viajero .....	56
Ilustración 74: Patrón Modelo-Vista-Controlador .....	57
Ilustración 75: Diagrama capa vista .....	58
Ilustración 76: Diagrama capa controlador .....	59
Ilustración 77: Diagrama capa modelo .....	59
Ilustración 78: Vista arquitectónica .....	60
Ilustración 79: Modelo de despliegue .....	60
Ilustración 80: Diagrama de secuencia añadir viajero diseño .....	61
Ilustración 81: Estructura general de la aplicación .....	62
Ilustración 82: Estructura carpeta Java .....	62
Ilustración 83: Menú principal de la aplicación .....	65
Ilustración 84: Gestión de viajeros 1 .....	66
Ilustración 85: Gestión de viajeros 2 .....	66
Ilustración 86: Eliminar viajero .....	67
Ilustración 87: Nuevo viajero 1 .....	68
Ilustración 88: Nuevo viajero 2 .....	68
Ilustración 89: Gestión de rueda sin horario .....	69
Ilustración 90: Gestión de rueda con horario generado .....	69
Ilustración 91: Horario ampliado .....	70
Ilustración 92: Compartir horario .....	71
Ilustración 93: Reiniciar rueda .....	72

## Resumen

En este proyecto, elaborado para el Trabajo de Fin de Máster del Máster en Ingeniería Informática de la Universidad de Salamanca, se pretende desarrollar una aplicación para dispositivos móviles con el sistema operativo Android que permita optimizar los viajes por trabajo.

Esta aplicación pretende optimizar y automatizar el proceso de elaboración de lo que en algunos gremios se conoce comúnmente como “la rueda”, que consiste en un horario utilizado por un grupo de trabajadores que residen en un mismo sitio y tienen un centro de trabajo común en un núcleo poblacional distinto o una ubicación lejana.

El horario se genera con la intención de que los trabajadores compartan sus vehículos en los desplazamientos, haciendo los viajes más seguros, amenos, económicos y respetuosos desde el punto de vista ecológico.

El calendario desarrollado debe tener en cuenta las siguientes particularidades:

- Todos los participantes que trabajen un día han de desplazarse al centro de trabajo y regresar del mismo en alguno de los vehículos movilizados.
- Es imperativo respetar el horario laboral de las personas desplazadas en cada vehículo, de tal manera que un participante que viaje en un coche nunca puede llegar al centro de trabajo a una hora posterior al comienzo de su horario laboral ni salir del centro antes del final de este.
- Se pretende que los trabajadores movilicen de manera semanal su vehículo particular un número máximo de veces, y que, si existe algún desequilibrio en la cantidad de desplazamientos, se compense económicaamente en proporción.
- Uno de los principios más complicados de efectuar consiste en la optimización del tiempo de los trabajadores desplazados, haciendo que el número de horas esperadas por estos en su centro de trabajo más allá de lo que marca estrictamente su horario laboral sean las menos posibles.
- Se intentará minimizar el número de vehículos desplazados al centro de trabajo, favoreciendo al grupo económico y generando un calendario más respetuoso con el medio ambiente.

El “**calendario de rueda**” semanal se suele realizar de forma manual, lo que quiere decir que, alguno de los integrantes del grupo de “la rueda” debe ser el encargado del desarrollo de este, invirtiendo una importante cantidad de su tiempo y esfuerzo en calcularlo.

Salta a la vista que un horario realizado de forma manual no tiene por qué representar el **resultado óptimo**, de hecho, es muy complicado que se obtenga el mejor resultado si no se utilizan herramientas computacionales.

Pero este trabajo no ha de realizarse una única vez, en ciertos tipos de trabajo resulta muy común que parte del elenco varíe cuantiosamente en cortos períodos de tiempo, siendo frecuente que aparezcan nuevos trabajadores o que algunos empleados finalicen contrato. Este tipo de cambios generan grandes alteraciones que invalidan el calendario ya realizado e implican la realización de otro nuevo, teniendo que repetir todo el desarrollo.

Para automatizar este proceso de elaboración, el software desarrollado ha de ser capaz de generar todos los calendarios posibles dados los viajeros y sus horarios, valorar cuál es el óptimo y representarlo en un **archivo exportable**.

## **Palabras clave**

La rueda, calendario de rueda, resultado óptimo, archivo exportable.

# Summary

This project, elaborated for the master's Thesis of the master's degree in Computer Engineering at the University of Salamanca, consists of the development of an application for mobile devices with the Android operating system to optimise business trips.

This application has the goal of optimising and automating the process of making what is commonly known in some guilds as "**la rueda**", a timetable used by a team of workers who reside in the same location and have a common workplace in a remote location.

The timetable is generated to enable workers to share their vehicles for travel, making journeys safer, more enjoyable, more economical, and more environmentally friendly.

The timetable developed should consider the following features:

- All participants working on a given day must travel to and from the workplace in one of the vehicles.
- It is mandatory to be in accordance with the work schedule of the people travelling in each vehicle, so a participant travelling in a car can never arrive at the workplace later than the start of his or her working hours or leave the workplace before the end of his or her working hours.
- The workers are expected to drive their private vehicle a maximum number of times per week and, if there is an imbalance in the number of drives, to be compensated financially in proportion.
- The time of the workers who have travelled in the vehicles will be optimised, so that the number of hours they spend at their workplace beyond their working hours should be as few as possible.
- It will try to minimise the number of vehicles transported to the workplace, benefiting the group economically and generating a more environmentally friendly calendar.

The weekly "**rueda calendar**" is usually done manually, which means that one of the members of the "rueda" group must oversee its development, spending a significant amount of time and effort in preparing the timetable.

A manual timetable does not necessarily represent the optimal result, in fact, it is very difficult to obtain the **best result** without the use of computational tools.

In certain types of work, it is very common for part of the work group to change considerably in short periods of time, with new workers appearing or some employees terminating their contracts. These changes cause major alterations that nullify the calendar that has been made and require the creation of a new one, which means that all the work must be repeated.

To automate this elaboration process, the software developed must be able to generate all possible calendars given the travellers and their schedules, evaluate which of the generated calendars is the optimal one and represent it in an **exportable file**.

## Keywords

La rueda, rueda calendar, best result, exportable file.

# 1. Introducción

Este Trabajo de Fin de Máster nace de una situación real, una necesidad que le ha surgido al alumno a lo largo de este año mientras se desempeñaba en su puesto de trabajo.

Es muy común, más aun, teniendo en cuenta el marco de trabajo actual, que a ciertas personas les surja la necesidad de desplazarse en coche a otro núcleo poblacional para poder asistir a su puesto de trabajo.

Esta situación se ve acentuada en ciertos gremios, como podría ser, entre otros, educación, la policía o sanidad, ya que sus puestos de trabajo son asignados por oposiciones, listas de elección y otros métodos de distribución que tienden a generar situaciones en las cuales el lugar de residencia y el centro de trabajo del individuo no coinciden físicamente en una misma ciudad o pueblo.

También puede generarse esta circunstancia en puestos de trabajo situados en lugares menos habitados, como pueblos o, incluso, fuera de núcleos poblacionales. Es valorable también el caso inverso, un individuo que decide residir en zonas menos habitadas es muy probable que tenga que desplazarse para poder ejercer su profesión.

Según un estudio de Michel Page (Page, s.f.), en el cual se analiza una muestra de más de 12.000 encuestados, el 39% de los europeos tarda más de 45 minutos en llegar al trabajo y el 66% de las personas encuestadas utilizan su transporte privado para realizar este desplazamiento. Haciendo uso de este mismo documento, en España, los trabajadores encuestados tardan una media de 36 minutos en llegar al lugar de trabajo.



Ilustración 1: Media de tiempo empleado para llegar al lugar de trabajo en Europa, estudio de Michael Page

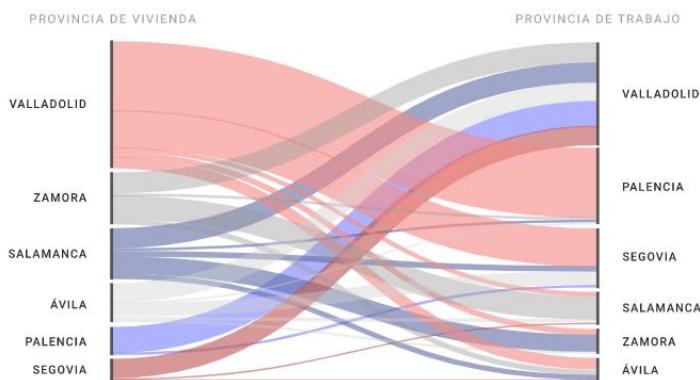
El 16 de febrero de 2020, en el periódico el Confidencial, se publicaba un artículo demoledor titulado “La gran migración diaria” en el cual se expone que 630.000 personas cambian de provincia a diario para ir a trabajar, representando este número al 3,2% de los ocupados. (Confidencial, s.f.)

Es decir, que a diario se ve en esta situación un volumen de población superior al de la ciudad de Málaga.

El aumento de este tipo de movilidad se asocia, entre otras cosas, a la mejora de las infraestructuras de transporte (que hace que estos desplazamientos sean posibles), al elevado precio de las viviendas en las grandes ciudades (que obligan a ciertas familias a vivir en ciudades donde el precio sea más asequible) a la existencia de jornadas semipresenciales (que permiten no asistir todos los días al lugar de trabajo).

Este estudio del Confidencial arroja datos y gráficas muy sorprendentes y relevantes, pero para este trabajo interesa especialmente la que se va a mostrar a continuación.

### PRINCIPALES MOVIMIENTOS INTERPROVINCIALES DE CASTILLA Y LEÓN



Fuente: EPA

*Ilustración 2: Principales movimientos interprovinciales de Castilla y León*

Se puede apreciar cómo, del mismo lugar de origen, se desplaza a diario un gran número de personas al mismo lugar de destino, y muy posiblemente al mismo lugar de trabajo.

En el caso del alumno, a diario, se desplazaban entre 6 y 8 personas desde Salamanca al mismo centro de trabajo en Benavente.

Desde hacía tiempo, estas personas llevaban haciendo una práctica muy conocida por los trabajadores que se ven obligados a desplazarse y que es conocida como “la rueda”.

Pero ¿Qué es “la rueda” y por qué mejora la vida de estos trabajadores?

“La rueda” simplemente se basa en la organización de estos trabajadores que han de desplazarse en sus coches particulares desarrollando un horario en el cual se tengan en cuenta los siguientes factores:

- **Distribución en coches particulares:** Se pretende que los participantes de “La Rueda” se desplacen al centro de trabajo o bien transportando o siendo transportados por compañeros. Puede darse la situación de que algún día en concreto un trabajador tenga que realizar algún viaje solo, aunque no es lo deseable.
- **La entrada y la salida de los trabajadores desplazados:** Los trabajadores nunca pueden llegar tarde a su puesto de trabajo ni pueden salir de este antes de que finalice su jornada laboral.
- **Equilibrio en los desplazamientos:** Se pretende que los trabajadores movilicen de manera semanal su vehículo particular un número máximo de veces, y que, si existe algún desequilibrio en la cantidad de desplazamientos, se compense económicaamente en proporción.
- **Espera mínima:** Este principio es el más complicado de ejecutar, ya que, teniendo en cuenta los principios anteriores y siendo conscientes de que todos los participantes de un día concreto han de llegar al centro de trabajo y volver del mismo, hay que generar un horario viable que, además, haga que los miembros de “la rueda” esperen la menor cantidad de tiempo posible en el centro de trabajo más allá de su jornada laboral.
- **Desplazamientos mínimos:** Pero claro, además de todos estos principios mencionados anteriormente cabe tener en cuenta que es conveniente desplazar la menor cantidad de coches posibles al centro de trabajo, haciendo el sistema más ecológico y económico.

Un buen “horario de rueda” mejora la vida de los trabajadores, haciendo que tengan que conducir menos horas y, por tanto, favoreciendo su descanso y seguridad, haciendo que tengan que esperar el menor tiempo posible en el centro de trabajo y proporcionando compañía en los desplazamientos. Todo esto hace los viajes más seguros, más económicos, más amenos y menos contaminantes.

Pero claro, esta organización tiene un problema y es que, de momento, el proceso de elaboración de este “horario de rueda” se viene haciendo a mano.

	Jueves.	Viernes.	Sáb. 11:00	Sáb. 14:00	Viernes.
1º	MARCOS. Núria Leti	ROBERTO CARLOS. Núria	LUCIA Marcus Núria	MARIA JESÚS Leti Maria. Núria.	NÚRICA MARCOS
2º	MARÍA Lucía.	LETI Nuria dúcia	A 3º LETI Nuria	Seven 11:00 LUCIA Marcus.	A 3º MARÍA Lucía.
5	MARÍA Núria Lucía. Leti.	LETI dúcia Nuria Núria.	LUCIA Núria	Seven A 14:00 Núria Leti Nuria	MARÍA Leti MARCOS
6	MARCUS.		LETI Nuria Marcus.	A 5º Nuria Jesús.	NÚRICA Nuria Jesú Lucía

Ilustración 3: Horario elaborado a mano

Obviamente, la elaboración manual del calendario requiere invertir bastante tiempo y esfuerzo para poder cuadrar los distintos horarios, siendo casi imposible hacer a mano el horario óptimo.

Además, en el trabajo del alumno fue muy común que aparecieran nuevos compañeros que se desplazaban desde Salamanca y querían participar en “la rueda”, así como bajas dentro de los participantes, ya fueran de cierto tiempo o permanentes. Todo esto provoca que el horario desarrollado deba recalcularse casi semanalmente.

La idea de este proyecto consiste en automatizar este proceso, de tal manera que los cálculos y la generación del horario se realice de manera automática, evitando así los problemas que puede generar la elaboración manual y maximizando los beneficios.

## 2. Objetivos del proyecto

El objetivo principal del proyecto es desarrollar un software que optimice los viajes por trabajo, generando calendarios óptimos, para que un grupo de personas que desean desplazarse de manera conjunta pueda realizarlo de una manera eficiente, respetando los horarios laborales de los viajeros, minimizando el tiempo que estos están innecesariamente en el centro de trabajo y reduciendo el número de coches desplazados semanalmente.

Este calendario tiene como objetivo mejorar estos desplazamientos, haciéndolos más económicos, más amenos, más seguros y mejorando la gestión de tiempos.

### 2.1. Objetivos funcionales

Los objetivos funcionales se corresponden con aquellos que describen las actividades que el sistema ha de llevar a cabo y los servicios o funciones que debe proporcionar, valorando las distintas entradas que puede recibir y el estado de este. (Blanco, s.f.)

#### 2.1.1. Gestión de viajeros:

El primer objetivo es que el usuario pueda añadir los datos de los participantes de “la rueda” en la aplicación, de tal manera que pueda modificar y eliminar esta información cuando deseé.

Estos datos se van a utilizar posteriormente para poder llevar a cabo los objetivos que se enunciarán a continuación.

#### 2.1.2. Calcular todos los “horarios de rueda” posibles

El software ha de calcular todos los “horarios de rueda” posibles, ateniéndose en todo momento a las restricciones ya indicadas en el apartado de introducción.

Los horarios generados indicarán, de todos los días de la semana, en qué coche se desplazarán los viajeros, tanto para la ida como para la vuelta y quién será el conductor.

#### 2.1.3. Evaluar todos los horarios de rueda generados

Se procede a evaluar todos los horarios generados, utilizando como atributos significativos el número de horas esperadas por todos los viajeros en el centro de trabajo (entendiendo por horas esperadas aquellas que van más allá del horario laboral del mismo) y el número de vehículos desplazados.

#### 2.1.4. Seleccionar el mejor resultado

Teniendo todos los atributos valorados toca ponderarlos y seleccionar el resultado considerado como óptimo.

#### 2.1.5. Generar un fichero exportable con el mejor resultado

Para permitir a los usuarios de la aplicación compartir de una forma sencilla el “calendario o horario de rueda” óptimo generado, el sistema ha de generar un fichero (preferiblemente PDF) con los datos del mejor resultado obtenido.

Este fichero ha de poder compartirse desde la aplicación utilizando apps de terceros.

### 2.1.6. Reiniciar el sistema

Debido a que pueden darse situaciones en las que cambia completamente el equipo de trabajo, se pretende proporcionar al usuario una manera de eliminar de manera eficaz toda la información almacenada en el sistema, devolviendo la aplicación a su estado inicial.

## 2.2. Objetivos no funcionales

Los objetivos no funcionales son aquellos que no describen actividades o funciones del sistema, están más asociados con restricciones del diseño o estándares de calidad.

### 2.2.1. Accesibilidad

Se pretende que el software sea accesible para la mayor cantidad de usuarios finales posibles y que, una vez tengan descargado e instalado el software, su uso no suponga una barrera.

### 2.2.2. Funcionamiento en Android

Con la intención de favorecer esta accesibilidad a la que se ha hecho mención previamente, se ha decidido que la aplicación se desarrolle para el sistema operativo Android.

El teléfono móvil ya es el dispositivo inteligente más utilizado según un estudio realizado por vpnMentor. Desde 2017 el móvil ha superado el uso del ordenador de sobremesa y mantiene una variación anual ascendente desde entonces (vpnmentor, s.f.).



Ilustración 4: Porcentaje de tráfico web por dispositivo en 2020

Según los datos publicados por Yi Min Shum Xie en 2020 el 67% de la población global tiene un dispositivo móvil, y de estos dispositivos el 74% se corresponde con el sistema operativo Android. (Xie, 2020)

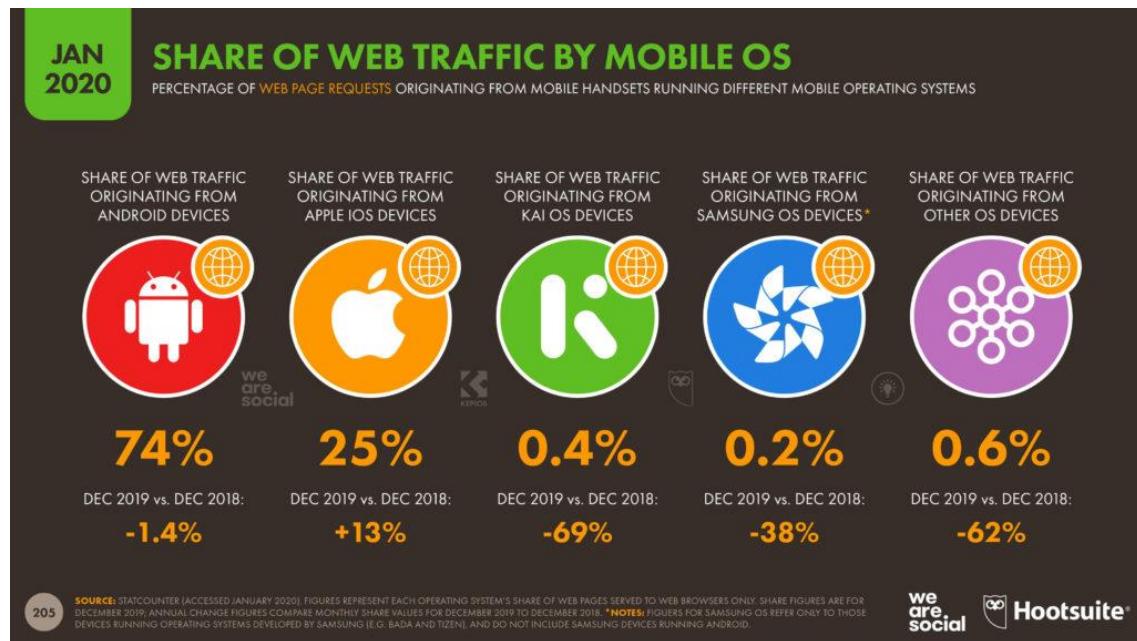


Ilustración 5: Sistemas operativos de móvil más utilizados

Por tanto, parece coherente programar para el sistema operativo más utilizado del dispositivo inteligente más accesible y utilizado.

#### 2.2.3. Adaptabilidad y escalabilidad

El tiempo de trabajo para este desarrollo es limitado, por tanto, el alumno se centrará en resolver el problema planteado de una manera eficaz, pero a lo largo de los siguientes años podrían añadirse nuevas funcionalidades o, incluso, exportar la aplicación desarrollada a otro tipo de dispositivos y sistemas operativos.

Por tanto, el desarrollo se llevará a cabo con la premisa de facilitar posibles mejoras futuras y la adaptabilidad de la aplicación a otro sistema.

#### 2.2.4. Interfaz amigable y sencilla

Debido a que el usuario final no tiene por qué poseer unos conocimientos elevados de informática ni del uso de dispositivos móviles, se pretende favorecer que el funcionamiento de la aplicación sea lo más intuitivo y sencillo posible, generando así una mayor cantidad de usuarios objetivo.

#### 2.2.5. Eficacia

Esta aplicación ha de competir directamente con la elaboración manual de estos mismos horarios.

Planificar un mal horario manualmente no es difícil, lo complicado es generar el horario óptimo, por tanto, se ha de priorizar la generación del mejor horario, aunque esto conlleve una penalización de tiempo en la ejecución del algoritmo.

### 3. Estudio de tecnologías similares

Una vez se ha tenido la idea que dará forma a este proyecto, llega el momento de analizar cuál será la competencia directa, si es que existe.

Como ya se mencionó anteriormente, la idea surge por la adhesión del alumno a un grupo que llevaba a cabo esta organización para el desplazamiento hasta el centro de trabajo y, tras consultar a los distintos integrantes (incluidos algunos que llevaban años realizando esta práctica) descubrió que ninguno conocía una solución informática para desarrollar el calendario deseado. Por tanto, o no existía competencia, o estaba orientada hacia otra solución del problema o la competencia existente no era demasiado utilizada.

Volviendo al estudio mencionado llevado a cabo por (Confidencial, s.f.) en la introducción, interesa analizar cómo ha crecido en los últimos años la cantidad de trabajadores que se ven obligados a cambiar de provincia para llegar a su puesto de trabajo, llegando a marcar un máximo en 2019



Fuente: EPA

Ilustración 6: Gráfica personas que cambian de provincia por trabajo en España

Esta información puede servir como indicador del público objetivo, que en número absoluto será mayor, ya que en muchos casos la aplicación será utilizada por personas que se desplazan para asistir a su centro de trabajo en la misma provincia, incluso ciudad en el caso de las grandes urbes.

Con esta gran cantidad de público objetivo, sorprende que no se conozca ninguna tecnología que permita elaborar calendarios para grupos de trabajo que se desplazan de manera conjunta. Se procederá a investigar más a fondo.

#### 3.1. Tecnologías similares

Ciertamente, tras un estudio en profundidad de las tecnologías que podrían estar relacionadas con este proyecto, se ha descubierto que no existiría ninguna competencia directa como tal, ninguna de las herramientas investigadas ofrece la oportunidad de generar horarios óptimos de viaje dadas las restricciones de horarios, más bien están orientadas tan solo al reparto económico y a la búsqueda de compañeros.

### 3.1.1. Dedocar



**App para compartir coche al trabajo**

Ahórrate tiempo, gastos y problemas a diario, compartiendo asientos libres con quienes van en tu misma dirección.

*Ilustración 7:Logo dedocar*

Dedocar presume en su página web de ser la aplicación de móvil que ayuda a sus usuarios a compartir coche para ir al trabajo.

Esta aplicación es una de las opciones más reconocidas para compartir vehículo en los desplazamientos por trabajo, tanto es así que si se realiza una búsqueda en Google introduciendo la frase “compartir vehículo para ir a trabajar” es la segunda opción que propone el buscador.

compartir vehículo para ir a trabajar

Todo Imágenes Videos Noticias Maps Más Herramientas

Aproximadamente 141.000.000 resultados (0.69 segundos)

<https://journify.com> ▾  
**Journify: App para Compartir Coche**  
Journify es la solución de carpooling inteligente para compartir coche a diario al trabajo. Comparte tus trayectos y muévete de forma sostenible y económica ... Visitaste esta página el 29/08/21.

<https://dedocar.org> ▾  
**App para compartir coche al trabajo – Ahórrate tiempo, gastos ...**  
DedoCar es la nueva aplicación móvil que te ayuda a compartir coche para ir al trabajo. ¿Estás harto de perder tanto tiempo en ir a trabajar, o estudiar? ¿O ... Cómo ir de pasajero · FAQs · Cómo ir de conductor · ¡Apúntate!  
Visitaste esta página el 29/08/21.

*Ilustración 8: Búsqueda compartir vehículo para ir a trabajar*

Realmente esta aplicación es una especie de “BlaBlaCar” que se encarga de poner en contacto a conductores que buscan viajeros y a viajeros que buscan conductor a través de un buscador en el que se especifican el tanto el origen como el destino.

Parece que la aplicación de móvil que ofrecen ha sido eliminada de Play Store.



Ilustración 9: Dedocar en Play Store

Este tipo de aplicaciones dependen mucho de tener una comunidad grande y muy activa (no es el caso) y además no atacan el problema desde el mismo punto de vista, dedocar entiende que el usuario aún no conoce a nadie que se desplace desde el mismo punto de origen que él en dirección al mismo punto de destino y pone en contacto a estos individuos, pero no soluciona la organización de estos trabajadores una vez están puestos en contacto, únicamente establece un precio al viaje y compensa económica mente al conductor. (dedocar, s.f.)

### 3.1.2. Journify

Si dedocar era el segundo resultado en la búsqueda Google con la frase “compartir vehículo para ir a trabajar” el primer resultado es Journify.



Ilustración 10: Búsqueda compartir vehículo para ir a trabajar

Journify es una startup valenciana que tiene como objetivo ofrecer un servicio similar al de BlaBlaCar, pero enfocado a trayectos diarios, un nicho que según ellos aseguran, no estaba cubierto. (Plaza, 2019)

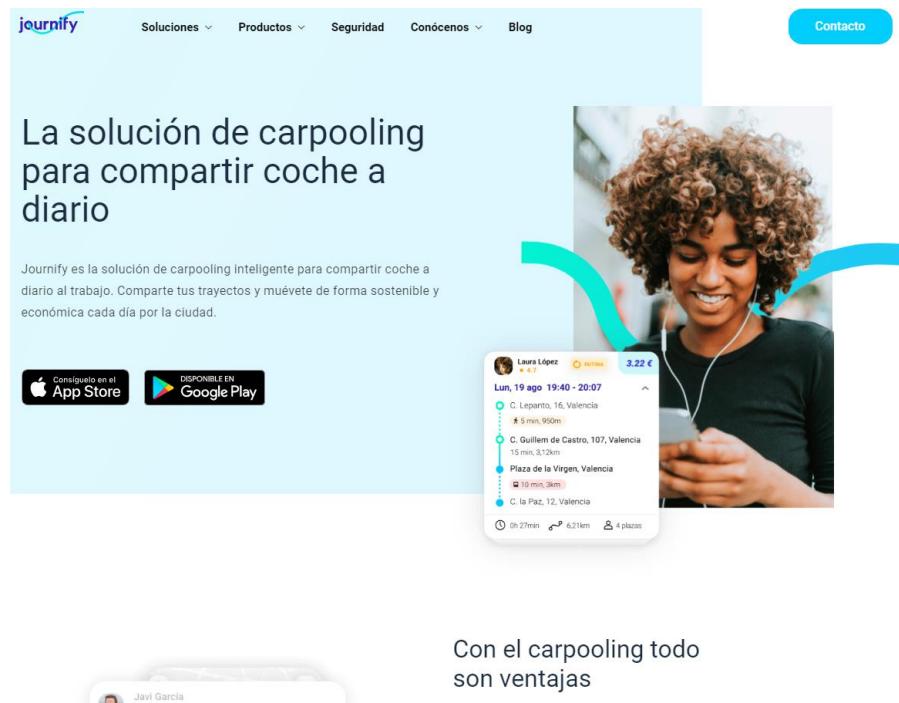


Ilustración 11: Interfaz journify

Esta aplicación solicita al conductor sus rutinas, establece un precio por asiento y el pasajero elige la opción que mejor le encaja.

La aplicación retiene entre un 10% y un 15% del precio correspondiente a cada asiento reservado.

El alumno ha intentado instalarla en un dispositivo iPhone y ha dado problemas.

Realmente sería una competencia directa de Dedocar más que de la aplicación desarrollada en este trabajo ya que la forma con la que atacan el problema es la misma. (journify, s.f.)

El alumno ha procedido a probarla en su dispositivo móvil:

Primero en iPhone.



Ilustración 12: Búsqueda Journify App Store

Una vez instalada se procede a probar su utilidad.

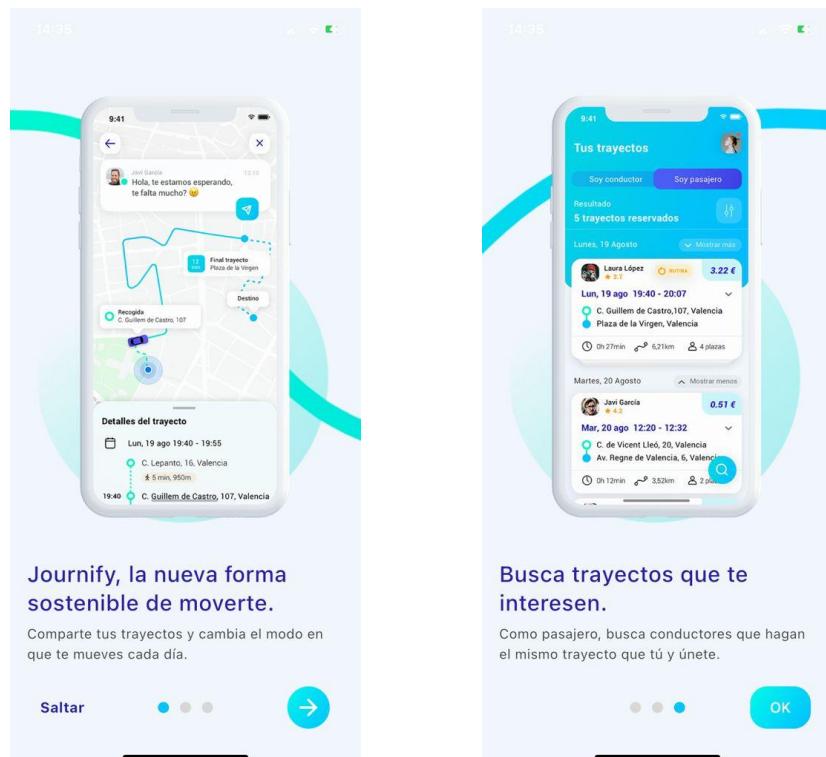


Ilustración 13: Interfaz inicial de Journify

Ilustración 14: Interfaz inicial de Journify

Presenta una interfaz atractiva.

Después de una introducción solicita el número al usuario.

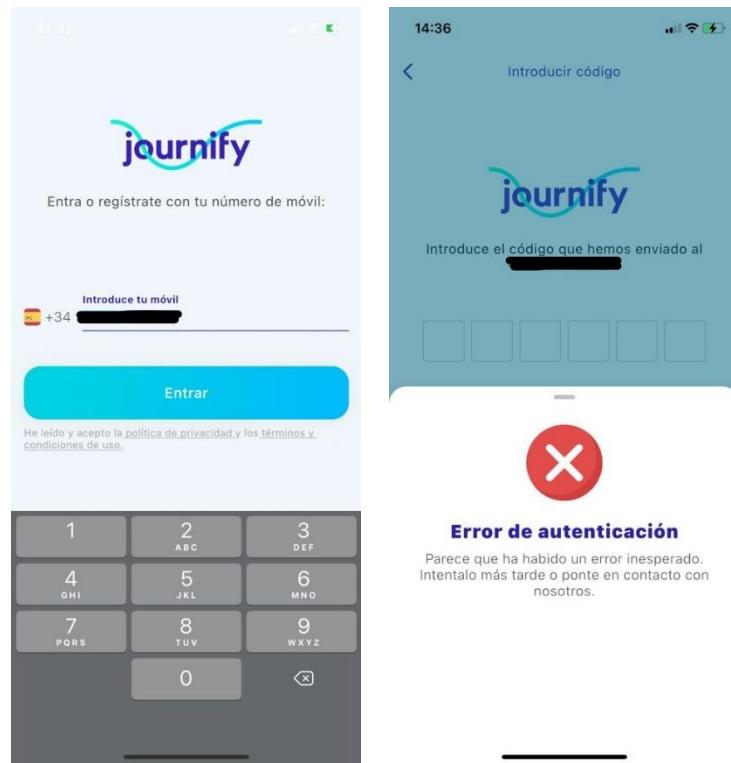


Ilustración 15: Solicitud de teléfono Journify

Ilustración 16: Error en IOS Journify

El alumno no ha podido acceder a la funcionalidad en la app de iPhone porque no funciona bien.

Este registro si funciona correctamente en la aplicación Android.

Se probó el funcionamiento de la aplicación para el viaje que realizaba el alumno.

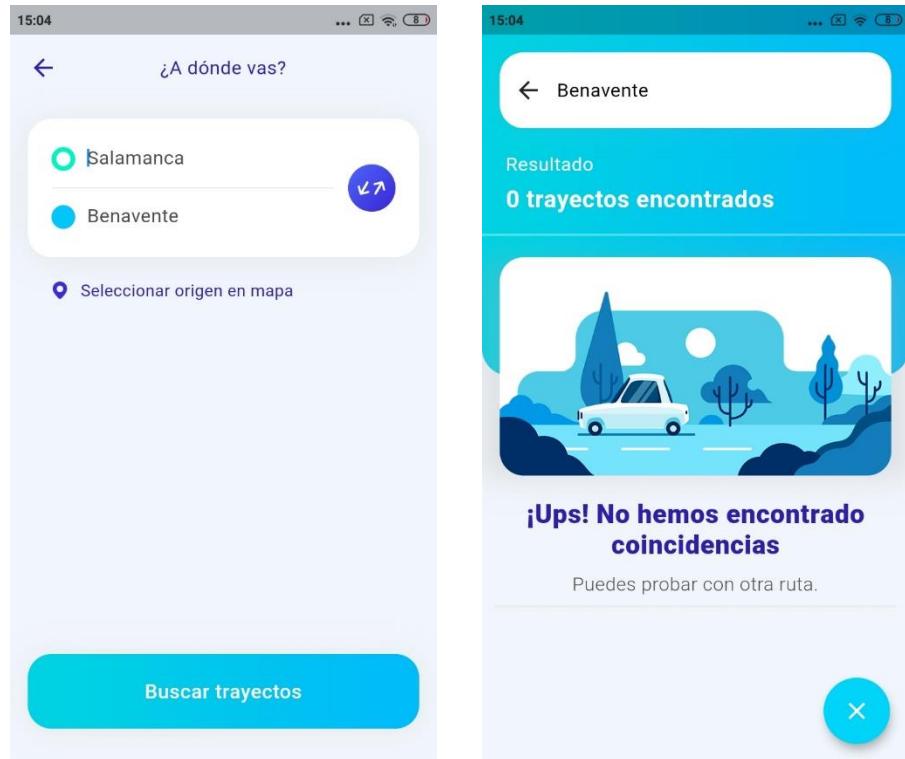


Ilustración 17: Búsqueda Salamanca-Benavente Journify

Ilustración 18: Búsqueda sin resultados

Pero no se encontró ningún resultado.

Se probó con rutas más usuales como Toledo-Madrid o Getafe-Madrid, pero no se obtuvo ningún resultado.

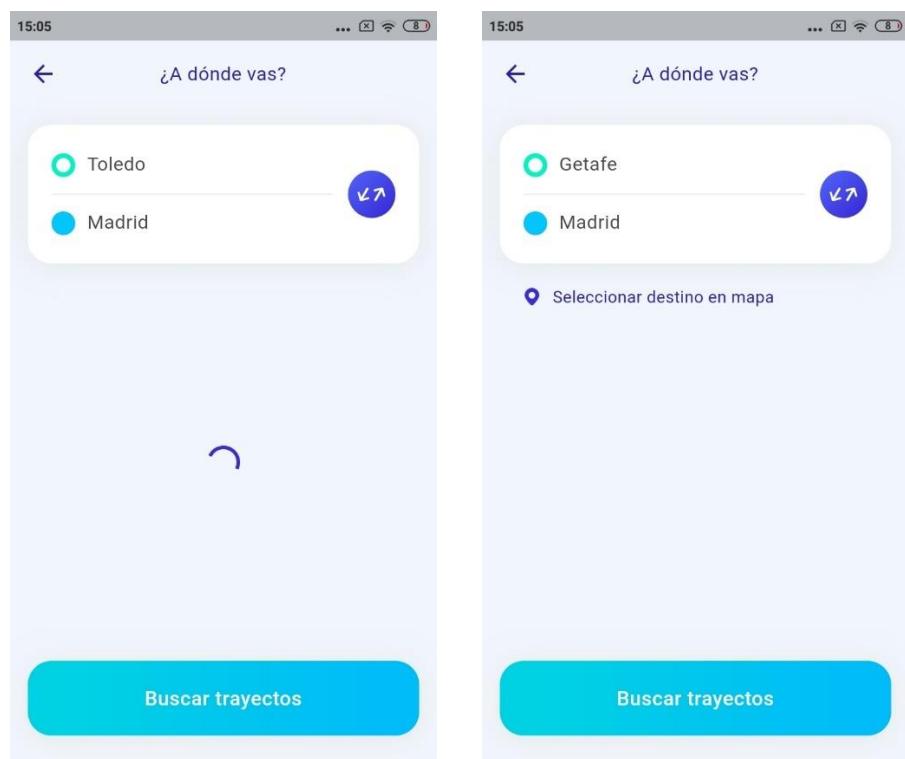


Ilustración 19: Búsqueda Toledo-Madrid Journify

Ilustración 20: Búsqueda Getafe-Madrid Journify

Se llega a la conclusión de que la comunidad no es activa y, por tanto, la funcionalidad de la aplicación es nula.

### 3.1.3. Carppling

Carppling confirma la tendencia mostrada en las tecnologías ya analizadas, y es que propone una solución al marco del problema desde un punto de vista económico, alguien ofrece un viaje, otra persona lo acepta y se llega a un acuerdo monetario. (carpling, s.f.)

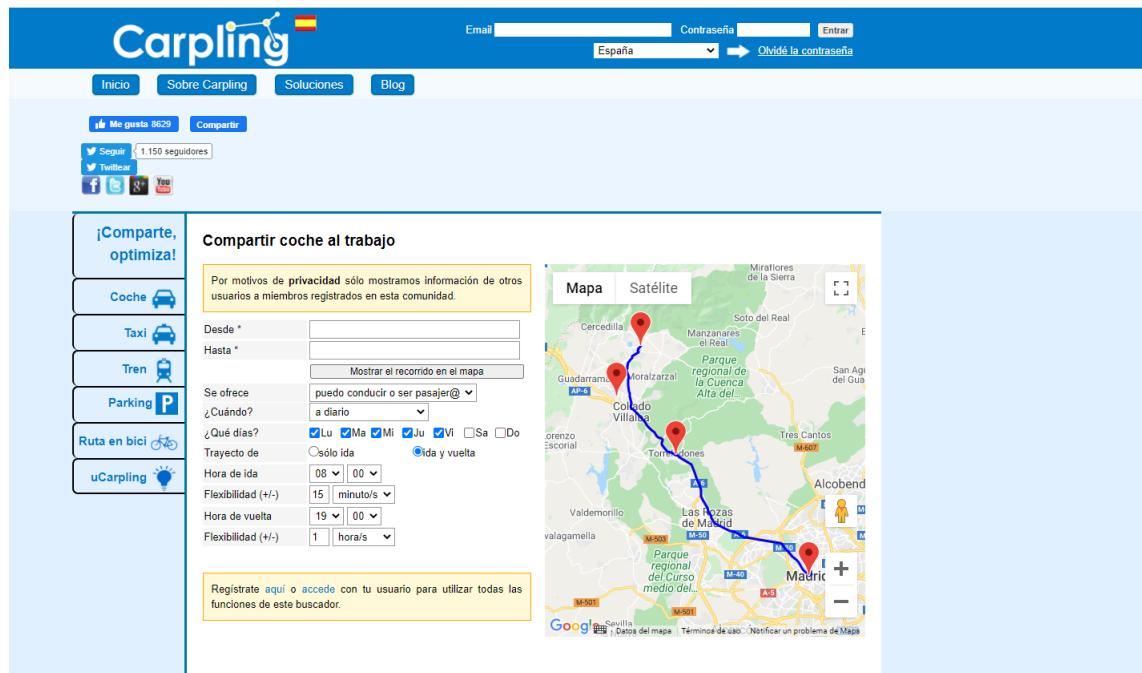


Ilustración 21: Interfaz Carpling

### 3.1.4. BlaBlaCar Daily

La empresa BlaBlaCar ha detectado un nicho de mercado en las aplicaciones orientadas a la organización de los viajes de trabajadores, no obstante, el proyecto que tienen pensado no saldrá a la luz hasta el año que viene.

Leyendo el artículo publicado por el periódico El Mundo (Montenegro, 2021) se puede interpretar que este gigante empresarial tiene pensado dar a su aplicación la misma orientación que la que tienen las apps antes mencionadas.

Es decir, será un funcionamiento similar al de BlaBlaCar, pero orientado a otro público objetivo, no está previsto que el software desarrollado realice cuadrantes u horarios.

**MEDIO AMBIENTE**

## Blablacar estrena servicio para compartir coche al trabajo

Pensado para viajes diarios de hasta 30 km ya opera en Francia y a España llegará en 2022. El servicio tradicional de Blablacar tiene seis millones de usuarios españoles



Adrien Tahon es el responsable de Blablacar Daily

*Ilustración 22: BlaBlaCar Daily*

### 3.2. Conclusiones

#### 3.2.1. Buen nicho de mercado

El nicho de mercado en el que busca hacerse hueco esta aplicación es real y ha sido detectado por varias empresas, incluso BlaBlaCar parece haber dedicado parte de sus recursos en la explotación de este público objetivo.

A pesar de que, claramente, numerosos desarrolladores han realizado ya alguna aplicación orientada a ayudar a los empleados que se desplazan a diario a su puesto de trabajo, no parece que ninguna haya calado en exceso en su público objetivo.

Dedocar parece haber caído definitivamente en desuso, ya que, su aplicación no se puede encontrar en Play Store.

Journify, aunque resulta la opción más cuidada y atractiva, tampoco parece que tenga un gran número de usuarios, ya que, encontrar otro trabajador con el que desplazarse es una misión imposible.

La app de BlaBlaCar Daily aún no está disponible en España.

Se considera, por tanto, que el nicho de mercado identificado es perfecto, ya que ha sido reconocido y tentado por empresas poderosas, pero aún no parece haber encontrado un producto de referencia al que agarrarse y, por tanto, el software desarrollado para contentar a estos usuarios aún está en pañales.

#### 3.2.2. Aplicación única

Pero más allá de que estas aplicaciones no hayan triunfado, ni siguiera orientan el problema de la misma manera que el software que se pretende desarrollar en este proyecto. Mientras las soluciones existentes buscan poner en contacto personas que se desplazan a diario desde un punto de origen a uno de destino, haciendo que una persona ejerza de taxista de otra mediante una compensación económica, este proyecto tiene un objetivo más ambicioso, que es generar un grupo de “rueda” en el cual todos ejerzan de conductores y pasajeros y mediante unos

cálculos, se consiga un horario óptimo en el que las prestaciones ofrecidas sean las mejores posibles.

Ha sido completamente imposible encontrar una aplicación parecida y que puede ser utilizada con este fin, por tanto, este proyecto será pionero.

### 3.2.3. Beneficios

Las aplicaciones consultadas identifican en este nicho de mercado un problema ambiental a resolver.

Desde el punto de vista de este proyecto, la posibilidad de realizar un “horario de rueda óptimo” repercutirá directamente en el medio ambiente, provocando que se desplacen menos vehículos y generando una sociedad más sostenible.

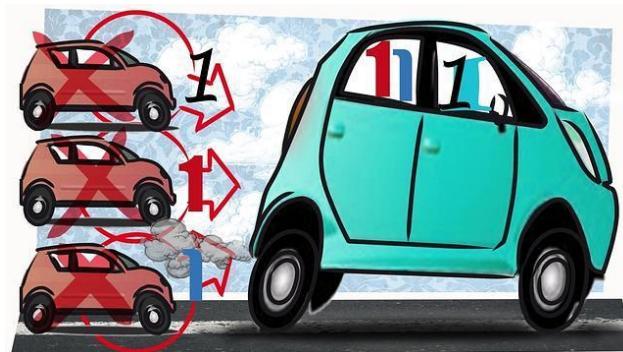


Ilustración 23: Reducción de vehículos

Pero es importante no perder de vista el objetivo principal, que reside en la comodidad del trabajador. Un buen “horario de rueda” hará que los desplazamientos sean más baratos, más amenos y repercutirá en una mayor cantidad de tiempo libre para el usuario.

## 4. Técnicas y herramientas

En este apartado se realiza un estudio sobre las herramientas utilizadas para la elaboración de este proyecto.

En un principio, se hizo una comparativa de las soluciones posibles para cada problema y se decidió que las herramientas y tecnologías que se expondrán a continuación eran las mejores.

### 4.1. Lenguajes de programación

#### 4.1.1. Java



Ilustración 24: Logo Java

Uno de los lenguajes de programación por antonomasia, su principal objetivo es que, una vez se ha desarrollado el programa, este se pueda ejecutar en cualquier plataforma.

El entorno de desarrollo integrado Android Studio, del que se hablará posteriormente, ofrece muchas herramientas que serán de ayuda para la programación de la aplicación móvil que se pretende desarrollar en el lenguaje java.

Por otra parte, en las distintas asignaturas de la universidad, así como en su trabajo, el alumno ha adquirido conocimientos necesarios para utilizar este lenguaje de programación, por tanto, el desarrollo de esta aplicación comenzará en un punto medio de la curva de aprendizaje.

Java siempre ha sido (y sigue siendo) uno de los lenguajes de programación más populares del mundo (statista, s.f.). Esto hace que exista una gran cantidad de información muy útil en todo desarrollo de un proyecto relativamente complicado.



Ilustración 25: Lenguajes de programación más usados

Todas estas particularidades han hecho decantar la balanza a favor de este lenguaje de programación orientado a objetos por encima de cualquier otro.

#### 4.1.2. SQL



Ilustración 26: Logo SQL

Durante el desarrollo de la aplicación, surgió la necesidad de almacenar los datos relativos a los trabajadores y sus horarios, para poder elaborar el “horario de rueda” en función de estos.

Android Studio da la posibilidad de introducir una pequeña base de datos SQLite en la aplicación, que es más que suficiente para poder almacenar la información deseada.

SQL es un lenguaje de consulta estructurado que permitirá manipular y descargar toda la información almacenada en una base de datos, así como generarla y es el lenguaje de facto utilizado para bases de datos. (scalegrid, s.f.)

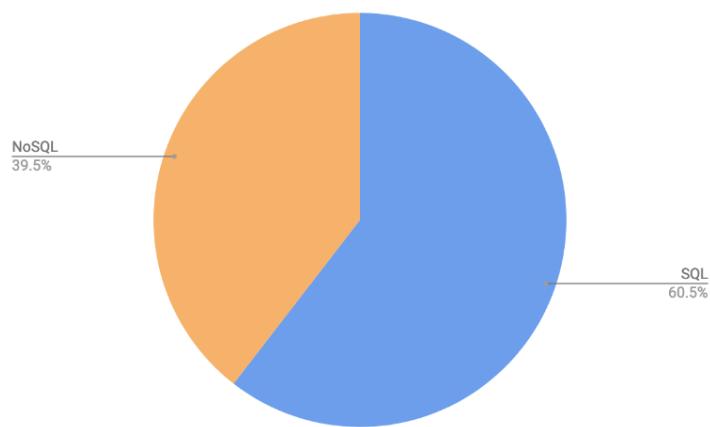


Ilustración 27: Porcentaje de uso SQL

La gran cantidad de usuarios que tiene este lenguaje, así como la formación impartida en la universidad al alumno sobre el mismo, ha facilitado enormemente la gestión de la base de datos desarrollada.

#### 4.1.3. XML



Ilustración 28: Logo XML

XML es un metalenguaje que permite definir lenguajes de marcado adecuados a usos determinados.

XML es el lenguaje elegido por Android Studio para que los desarrolladores realicen las interfaces de las aplicaciones móviles ya que este lenguaje es muy liviano y hace que el diseño no sea tan pesado.

Además de para vistas se utiliza para el manifest (archivo en el que se definen todos los componentes de la aplicación), archivos con textos reutilizables, archivos con colores reutilizables o archivos con estilos (entre otros).

## 4.2. Herramientas

### 4.2.1. Android Studio



Ilustración 29: Logo Android Studio

Es considerado el mejor Entorno de Desarrollo Integrado por los desarrolladores y por Google, que lo considera su software oficial, para el desarrollo de aplicaciones móviles para dispositivos Android.

Contar con el apoyo de Google le permite tener las mejores prestaciones en cuanto al acceso a funciones del sistema operativo, rendimiento y compatibilidad.

Android Studio también tiene las ventajas que puede presentar cualquier entorno popular: mucha documentación, recursos y soluciones.

El alumno ha cursado una optativa del máster sobre el desarrollo de aplicaciones móviles en la que se han impartido nociones sobre el uso de este entorno, muy útiles a la hora de llevar a cabo el proyecto.

Sobre este IDE se ha desarrollado la aplicación. (Esteban, 2020)

### 4.2.2. Gradle



Ilustración 30: Logo Gradle

Gradle se trata de una herramienta que permite la automatización de compilación de código abierto y se encuentra centrada en la flexibilidad y el rendimiento.

Cuenta con un sistema de gestión de dependencias muy estable.

Se apoya en Groovy y en un DSL para disponer de un lenguaje sencillo y claro a la hora de construir el build (en comparación con Maven).

Es el sistema oficial de compilación para Android y sus ficheros se autogeneran en Android Studio. (Muradas, s.f.) (Caules, 2015)

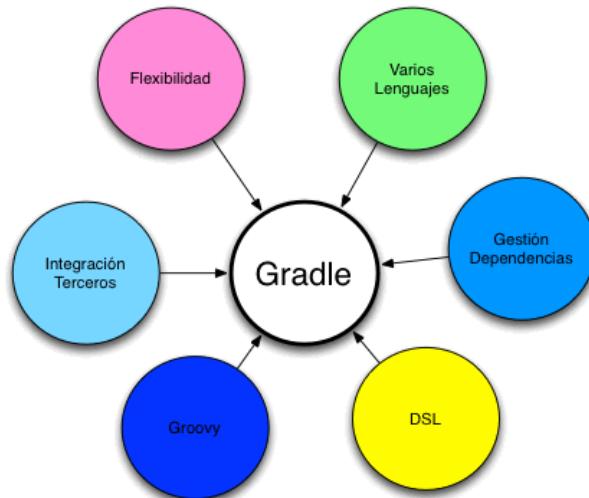


Ilustración 31: Funcionamiento Gradle

#### 4.2.3. Androidx



Ilustración 32: Logo AndroidX

AndroidX es el nombre resumido de “Android Extension Libraries” y realmente consiste en una nueva estructura de los paquetes Android que remplaza por completo a la antigua biblioteca de compatibilidad.

La jerarquía Android será tan solo para las clases predeterminadas de Android y otras bibliotecas serán parte de Androidx, provocando que los paquetes de Androidx se mantengan y actualicen por separado, sin necesidad de realizar ninguna operación sobre los paquetes de Android. (miReal, s.f.) (qastack, s.f.) (Android, <https://developer.android.com>, s.f.)



Ilustración 33: Explicación AndroidX

#### 4.2.4. Github



Ilustración 34: Logo GitHub

GitHub se trata de un portal diseñado para alojar el código de las aplicaciones de cualquier desarrollador, tal fue su éxito que en junio de 2018 fue comprada por Microsoft.

Ofrece muchas funcionalidades, que, además, van aumentando con el paso del tiempo, no obstante, en este caso, se ha trabajado siempre desde el mismo ordenador, y, el desarrollo, ha sido llevado a cabo únicamente por el alumno, por tanto, GitHub se ha utilizado como repositorio de confianza en el que almacenar el código en la nube. (xataka, s.f.)

#### 4.2.5. Android-pdf-viewer

Se trata de un repositorio público alojado en GitHub que ha resultado especialmente útil a la hora de realizar el proyecto.

Se trata de una biblioteca que permite visualizar documentos PDF en las aplicaciones de Android incluyendo animaciones, gestos, zoom y soporte para el doble clic.

Está basada en PdfiumAndroid para decodificar los archivos PDF.

Incluye una buena guía para su uso e instalación. (barteksc, s.f.)

#### 4.2.6. SQLite



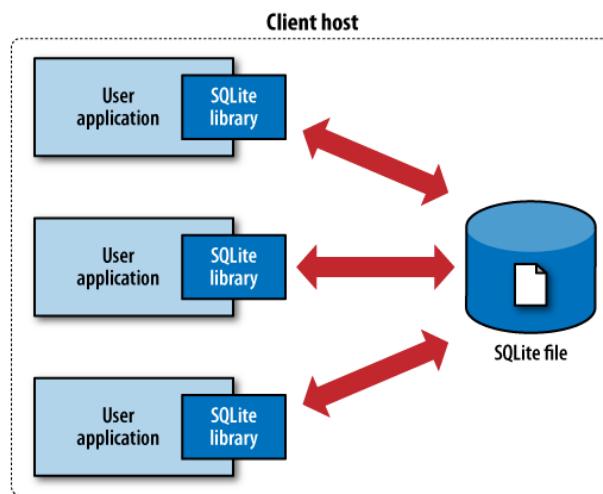
*Ilustración 35: Logo SQLite*

Se trata de un sistema de gestión de base de datos relacional de software libre, contenida en una biblioteca muy pequeña escrita en C.

Se diferencia de otros sistemas de gestión de bases de datos en que SQLite no se trata de un proceso independiente del programa principal, sino que se encuentra embebido en el programa formando parte integral del mismo y da lugar a una solución sencilla, eficaz, potente y rápida que puede empotrarse en dispositivos con pocas capacidades.

Es una opción muy recomendada para aplicaciones pequeñas que manejen bases de datos que no necesiten actualizarse por varios usuarios y que no sean excesivamente grandes.

Se puede instalar y utilizar de una manera relativamente sencilla en Android Studio, gestionando las bases de datos con el lenguaje SQL. (Rómmel)



*Ilustración 36: Funcionamiento SQLite*

#### 4.2.7. AVD Manager

Se trata de una interfaz, ofrecida por Android Studio, que permite administrar dispositivos virtuales que emulan aquellos basados en Android.

Por tanto, es uno de los métodos que ofrece el entorno de desarrollo utilizado para probar la aplicación desarrollada en una gran variedad de teléfonos, tabletas, relojes o televisión, dando la opción de elegir la versión de Android a utilizar. (androidcurso) (Ybrceyo)

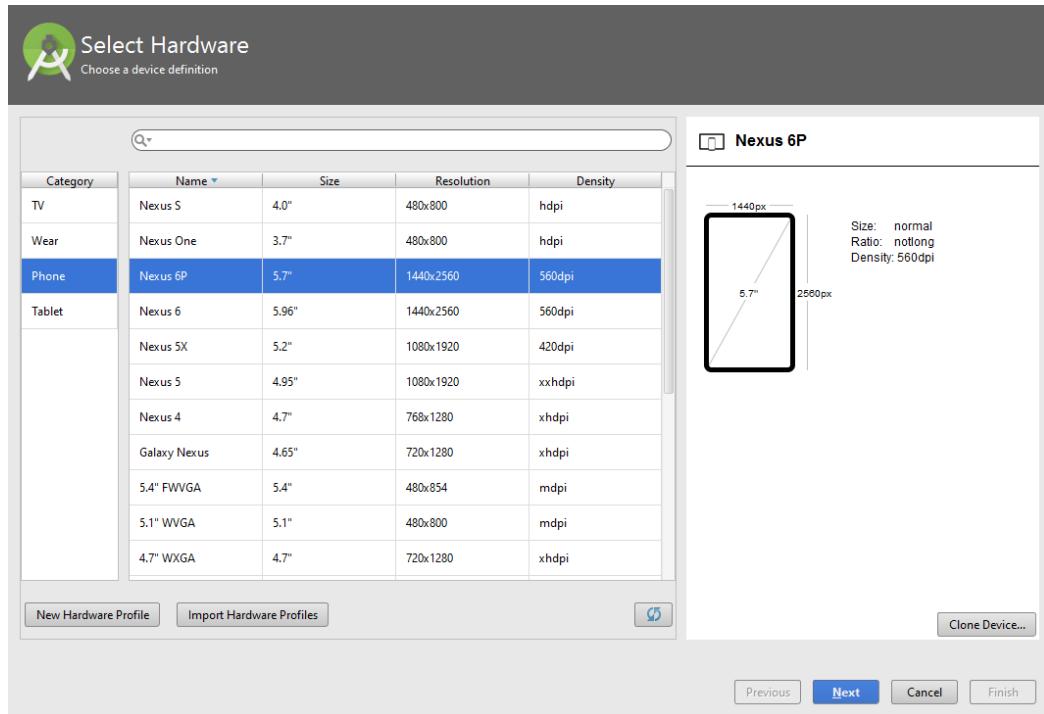


Ilustración 37: Hardwares simulables del AVD Manager

Ha sido la tecnología utilizada para probar de manera rápida la aplicación desarrollada para este proyecto.

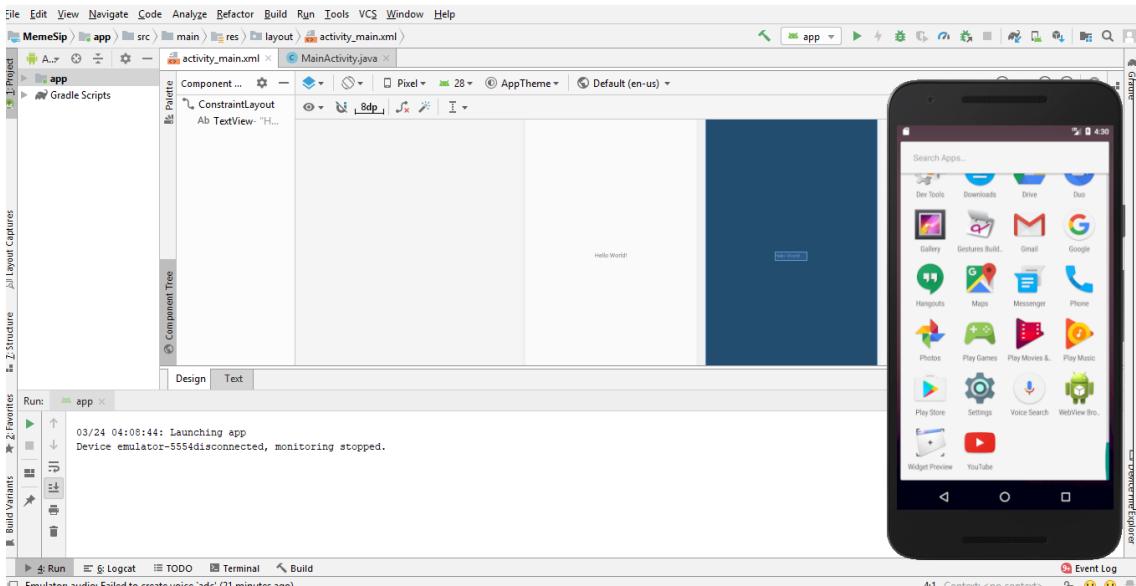


Ilustración 38: Teléfono simulado

También se ha probado la aplicación en un dispositivo móvil real.

#### 4.2.8. Microsoft Word



Ilustración 39: Logo Microsoft Word

Microsoft Word es una famosa aplicación ofimática, en concreto, un procesador de texto, y ha sido la herramienta elegida para realizar la memoria del proyecto y documentarlo.

El trabajo desarrollado tiene unas características muy concretas, en las que se solicita que el documento entregado se ciña a ciertas restricciones y estándares. En este caso Microsoft Word era el programa que mejor se adaptaba a esas particularidades.

Por otra parte, la popularidad de este software, hace que exista más documentación en internet, que puede ser especialmente útil para ciertos momentos en los que hay que utilizar funciones no tan conocidas de la aplicación.

#### 4.2.9. Visual Paradigm



Ilustración 40: Logo Visual Paradigm

Según su página web, se trata de una herramienta desarrollada para que el usuario pueda dibujar todo tipo de diagramas. Contiene un potente editor y un espacio de trabajo central que permite acceder y compartir el trabajo realizado.

Esta herramienta fue recomendada por los profesores de Ingeniería del Software de la Universidad de Salamanca en los años que el alumno cursó dicha carrera, y se impartieron nociones sobre el uso de esta tecnología.

Es de pago, no obstante, ofrece versiones gratuitas que, aunque presenten alguna limitación, han permitido llevar a cabo el trabajo que el alumno deseaba realizar perfectamente.

Ha sido el programa utilizado para llevar a cabo los diagramas UML que se pueden ver a lo largo de la memoria, y que se han realizado en las distintas fases por las que ha pasado el proyecto.

#### 4.2.10. EZEstimate

Se trata de una herramienta cuya función consiste en estimar el esfuerzo que requiere un proyecto a partir de sus casos de uso, de sus actores, de sus factores de complejidad técnica y de los factores de complejidad del entorno.

Tanto la descarga como el uso de esta herramienta se pudo llevar a cabo gracias a los conocimientos y recursos obtenidos en la asignatura Gestión de Proyectos.

Fue la herramienta usada para realizar la estimación de esfuerzo en el Anexo 1.

#### 4.2.11. Microsoft Project



Ilustración 41: Logo Microsoft Project

Se trata de un software de administración de proyectos desarrollado y vendido por Microsoft, siendo esta la herramienta más usada a nivel mundial para gestionar proyectos.

Permite, entre otras muchas funciones, administrar proyectos y gestionar la asignación de tareas y recursos, por tanto, ha sido utilizada en el Anexo 1 en el apartado de planificación temporal, ayudando a seguir los principios del Proceso Unificado y generando gráficos al estilo de diagramas de Gantt. (esan, 2018)

## 5. Aspectos relevantes del desarrollo

Se hará un resumen de los aspectos más importantes de los Anexos que recogen las fases de desarrollo del proyecto.

### 5.1. Marco de trabajo

Se ha decidido utilizar como modelo de desarrollo software el Proceso Unificado.

Está basado en componentes e interfaces bien definidas, dirigido por casos de uso, es iterativo e incremental y cada ciclo de desarrollo concluye con una versión entregable del producto. (Calabar, s.f.)

Los desarrollos software que utilizan este modelo pasan por cuatro fases:

- **Inicio:** Se define el alcance del proyecto, los objetivos de este, se analizan los factores de riesgo y se desarrollan los casos de uso.  
Es la única fase que no necesariamente genera una versión ejecutable.
- **Elaboración:** Finaliza el análisis de los casos de uso y se define la arquitectura del sistema.
- **Construcción:** Se construye la mayor parte del producto.

Se compone generalmente por un ciclo de varias iteraciones, en las que se va realizando progresivamente la programación de los casos de uso. Este enfoque permite tener en etapas tempranas del desarrollo versiones del sistema.

- **Transición:** Se obtiene la versión beta del software, se corrigen problemas (en el caso de que existan y se hayan detectado) y se añaden las mejoras sugeridas en la revisión previa.  
Esta etapa finaliza cuando con el sistema llega a la fase de producción.

Las iteraciones del proceso unificado consisten en intervalos recurrentes en el tiempo en los que se irán repitiendo procesos incrementales. Cada iteración se divide en varias fases, las cuales son (entre otras): Modelado del negocio, requisitos, análisis, diseño, implementación y pruebas.

Cada una de estas fases constará de un conjunto de tareas.

En los hitos se evaluará el cumplimiento de tareas y objetivos.

**Hitos:** Los hitos pueden definirse como revisiones periódicas (se realizan al final de cada iteración) en las que se analiza el progreso llevado a cabo.

La aparición de un hito suele significar el fin de una iteración y el inicio de la siguiente.

(ima, s.f.) (Grial)

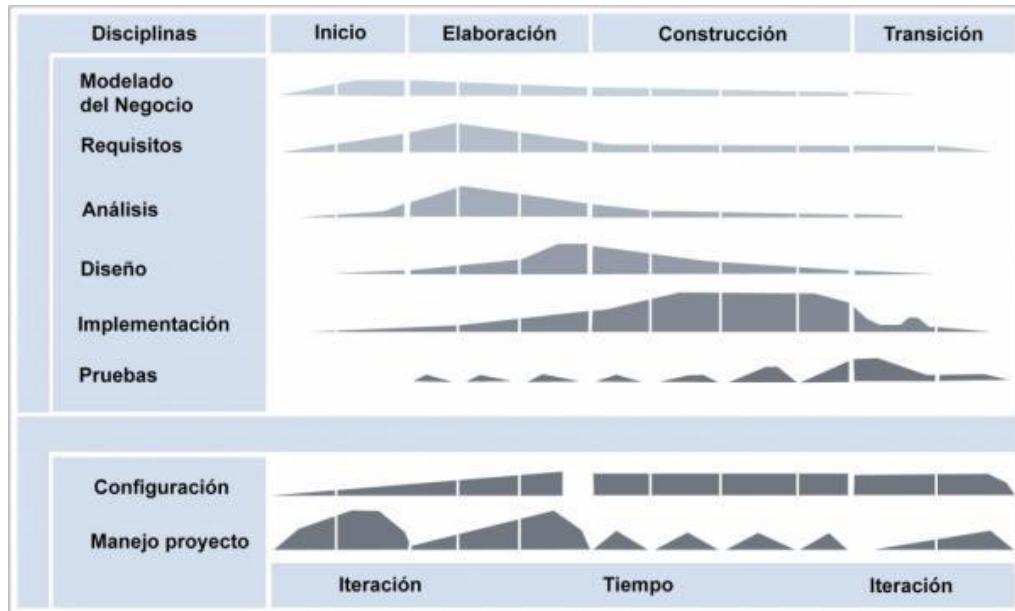


Ilustración 42: Esquema proceso unificado

## 5.2. Planificación temporal

En esta sección, se pretende definir las tareas que tendrán lugar en cada fase, siendo estas las propias de la metodología elegida (proceso unificado).

Cada fase se podrá desglosar en una serie de iteraciones, dentro de las cuales se pueden encontrar varias disciplinas (Modelado de negocio, requisitos, análisis, diseño, implementación y pruebas entre otras).

Dentro de cada disciplina se realizarán una serie de tareas, sobre las que habrá que indicar el recurso encargado (en este caso, al haber un solo recurso, será el encargado de todas las tareas) y la previsión de duración correspondiente a cada tarea.

Para cada iteración, se mostrará una captura con todas las tareas (en la que se puede ver correctamente los textos) y una que además incluye el diagrama de Gantt de la propia iteración en el que se podrá obtener una visión general de las tareas programadas (en estas capturas el texto se ve más pequeño, por eso se acompaña de las primeras capturas).

La información que se procede a mostrar a continuación es un resumen de la contenida en el Anexo 1, donde se profundiza sobre más sobre la planificación temporal.

### 5.2.1. Fase inicial

En esta primera fase, se realizó una reunión inicial con el tutor, en la que el alumno propuso por primera vez la idea de desarrollar lo que sería este proyecto, esta idea obtuvo la aprobación y se llevaron a cabo los trámites pertinentes de cara a oficializar su solicitud para presentarla como proyecto de Trabajo de Fin de Máster.

Una vez el director del máster aceptó la idea, se procedió a dar comienzo al trabajo, analizando los requisitos, instalando el software necesario para realizar diagramas, identificando los objetivos, definiendo los actores, definiendo los casos de uso y comenzando ligeramente el análisis.

Todas estas actividades han sido iterativas y valoradas en reuniones periódicas con el tutor, de tal manera que, el producto que se iba generando se refinaba en cada iteración.

	<b>Modo de tarea</b>	<b>Nombre de tarea</b>	<b>Duración</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Predecesoras</b>
1	➡	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
2	➡	▷ Iteración 1	3 días	lun 01/02/21	mié 03/02/21	
3	➡	▷ Modelado de negocio	2 días	lun 01/02/21	mar 02/02/21	
4	➡	Reunión inicial con el tutor	1 día	lun 01/02/21	lun 01/02/21	
5	➡	Estudio de tecnologías similares	1 día	mar 02/02/21	mar 02/02/21	4
6	➡	▷ Requisitos	1 día	mié 03/02/21	mié 03/02/21	3
7	➡	Establecimiento de los requisitos iniciales	1 día	mié 03/02/21	mié 03/02/21	
8	➡	Fin iteración 1	0 días	mié 03/02/21	mié 03/02/21	6
9	➡	▷ Iteración 2	8 días	jue 04/02/21	lun 15/02/21	8
23	➡	▷ Iteración 3	5 días	mar 16/02/21	lun 22/02/21	22
35	➡	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	1
93	➡	▷ Construcción	56,25 días	mié 07/04/21	jue 24/06/21	35
169	➡	▷ Transición	8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 43: Inicio, Iteración 1

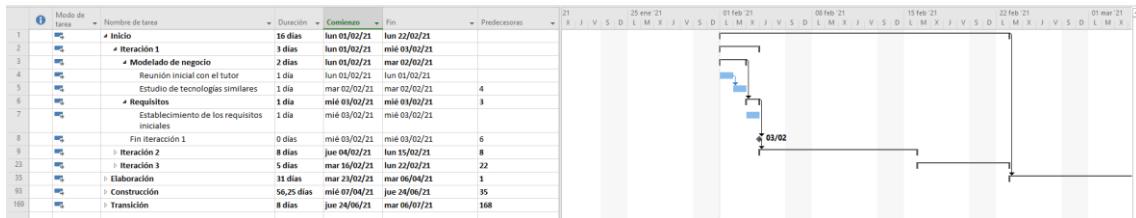


Ilustración 44: Inicio, Iteración 1 Gnatt

	<b>Modo de tarea</b>	<b>Nombre de tarea</b>	<b>Duración</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Predecesoras</b>
1	➡	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
2	➡	▷ Iteración 1	3 días	lun 01/02/21	mié 03/02/21	
3	➡	▷ Iteración 2	8 días	jue 04/02/21	lun 15/02/21	8
4	➡	▷ Modelado de negocio	4 días	jue 04/02/21	mar 09/02/21	
	➡	Nueva reunión con el tutor para analizar la viabilidad	1 día	jue 04/02/21	jue 04/02/21	
	➡	Envío de la solicitud de TFM al director del máster y espera de respuesta	3 días	vie 05/02/21	mar 09/02/21	11
	➡	Requisitos	2 días	mié 10/02/21	jue 11/02/21	10
	➡	Identificación de participantes y análisis de los mismos	0,2 días	mié 10/02/21	mié 10/02/21	
	➡	Obtenición de requisitos	1,3 días	mié 10/02/21	jue 11/02/21	14
	➡	Estudio de documentación	0,8 días	mié 10/02/21	mié 10/02/21	14
	➡	Observación	0,5 días	jue 11/02/21	jue 11/02/21	16
	➡	Refinamiento de los requisitos iniciales	0,5 días	jue 11/02/21	jue 11/02/21	15
	➡	Preparación de herramientas UML	2 días	vie 12/02/21	lun 15/02/21	13
	➡	Descarga e instalación herramientas UML	1 día	vie 12/02/21	vie 12/02/21	
	➡	Práctica herramientas UML	1 día	lun 15/02/21	lun 15/02/21	20
	➡	Fin Iteración 2	0 días	lun 15/02/21	lun 15/02/21	19
	➡	▷ Iteración 3	5 días	mar 16/02/21	lun 22/02/21	22
	➡	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	1
	➡	▷ Construcción	56,25 días	mié 07/04/21	jue 24/06/21	35
	➡	▷ Transición	8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 45: Inicio, Iteración 2

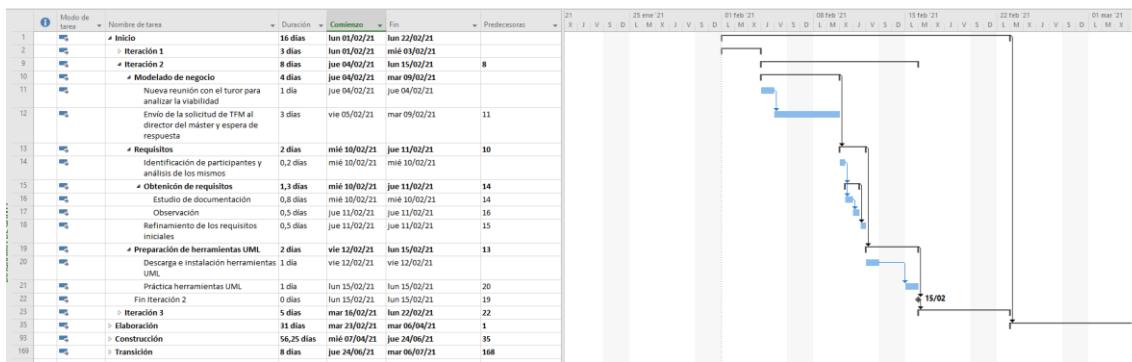


Ilustración 46: Inicio, Iteración 2 Gantt

	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	➡	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
2	➡	▷ Iteración 1	3 días	lun 01/02/21	mié 03/02/21	
9	➡	▷ Iteración 2	8 días	jue 04/02/21	lun 15/02/21	8
23	➡	▷ Iteración 3	5 días	mar 16/02/21	lun 22/02/21	22
24	➡	▷ Modelado de negocio	1 día	mar 16/02/21	mar 16/02/21	
25	➡	Reunión con tutor para explicar los progresos y la situación actual de los requisitos	1 día	mar 16/02/21	mar 16/02/21	
26	➡	▷ Requeritos	3,5 días	mié 17/02/21	lun 22/02/21	24
27	➡	▷ Obtención de requisitos	1 día	mié 17/02/21	mié 17/02/21	
28	➡	Observación	0,5 días	mié 17/02/21	mié 17/02/21	
29	➡	Entrevistas	0,5 días	mié 17/02/21	mié 17/02/21	28
30	➡	Definición de objetivos del sistema	0,5 días	jue 18/02/21	jue 18/02/21	27
31	➡	Generación de diagramas de casos de uso	2 días	jue 18/02/21	lun 22/02/21	30
32	➡	▷ Análisis	0,5 días	lun 22/02/21	lun 22/02/21	26
33	➡	Modelo de dominio	0,5 días	lun 22/02/21	lun 22/02/21	
34	➡	Fin Iteración 3	0 días	lun 22/02/21	lun 22/02/21	32
35	➡	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	1
93	➡	▷ Construcción	56,25 días	mar 06/04/21	jue 24/06/21	35
169	➡	▷ Transición	8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 47: Inicio, Iteración 3

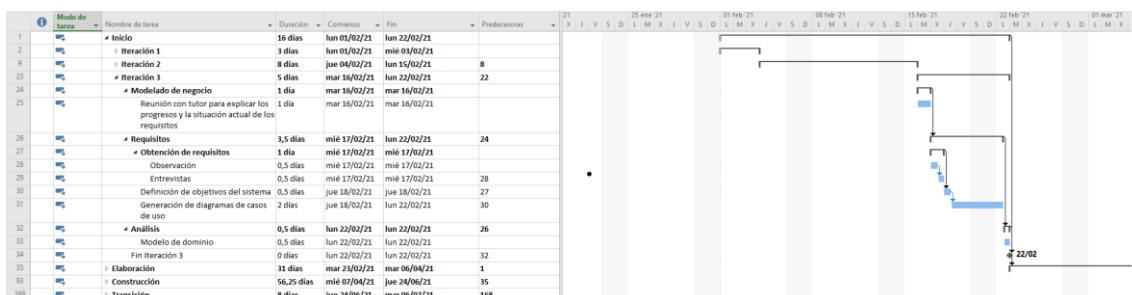


Ilustración 48: Inicio, Iteración 3 Gantt

## 5.2.2. Fase de elaboración

En la fase de elaboración, se han obtenido versiones casi definitivas de los casos de uso y sus diagramas, se ha realizado el diagrama de paquetes, los diagramas de comunicación, se ha desarrollado la vista de arquitectura, se han dibujado los diagramas de secuencia correspondientes a cada caso de uso (con una capa de abstracción alta), se han realizado los diagramas de diseño y una pequeña aproximación de los diagramas de secuencia correspondientes a la etapa de diseño que pertenecen a una capa de abstracción más baja que los ya mencionados.

Para culminar ha comenzado el desarrollo en Android Studio de la aplicación, programando las primeras funcionalidades.

Todas estas actividades han sido iterativas y valoradas en reuniones periódicas con el tutor, de tal manera que el producto que se iba generando se refinaba en cada iteración.

	<b>i</b>	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		➡ Inicio		16 días	lun 01/02/21	lun 22/02/21	
35		➡ Elaboración		31 días	mar 23/02/21	mar 06/04/21	1
36		➡ Iteración 1		8 días	mar 23/02/21	jue 04/03/21	
37		➡ Modelo de negocio		0,25 días	mar 23/02/21	mar 23/02/21	
38		Reunión con el tutor		0,25 días	mar 23/02/21	mar 23/02/21	
39		➡ Requisitos		3,75 días	mar 23/02/21	vie 26/02/21	37
40		Refinamiento de requisitos		0,25 días	mar 23/02/21	mar 23/02/21	
41		Obtención de requisitos de información		0,5 días	mar 23/02/21	mar 23/02/21	40
42		Refinamiento de los diagramas de casos de uso		2 días	mié 24/02/21	jue 25/02/21	41
43		Elaboración de los requisitos no funcionales		1 día	vie 26/02/21	vie 26/02/21	42
44		➡ Análisis		2 días	lun 01/03/21	mar 02/03/21	39
45		Refinamiento del modelo del dominio		1 día	lun 01/03/21	lun 01/03/21	
46		Elaboración del diagrama de paquetes		1 día	mar 02/03/21	mar 02/03/21	45
47		➡ Diseño		2 días	mié 03/03/21	jue 04/03/21	44
48		Estudio de posibles tecnologías		0,4 días	mié 03/03/21	mié 03/03/21	
49		Estudio de posibles patrones		0,4 días	mié 03/03/21	mié 03/03/21	48
50		Decisión de patrón y tecnologías a usar		0,2 días	mié 03/03/21	mié 03/03/21	49
51		Instalación de las tecnologías a usar		1 día	jue 04/03/21	jue 04/03/21	50
52		Fin Iteración 1		0 días	jue 04/03/21	jue 04/03/21	47
53		➡ Iteración 2		9 días	vie 05/03/21	mié 17/03/21	52
71		➡ Iteración 3		14 días	jue 18/03/21	mar 06/04/21	70
93		➡ Construcción		56,25 días	mié 07/04/21	jue 24/06/21	35
169		➡ Transición		8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 49: Elaboración, Iteración 1

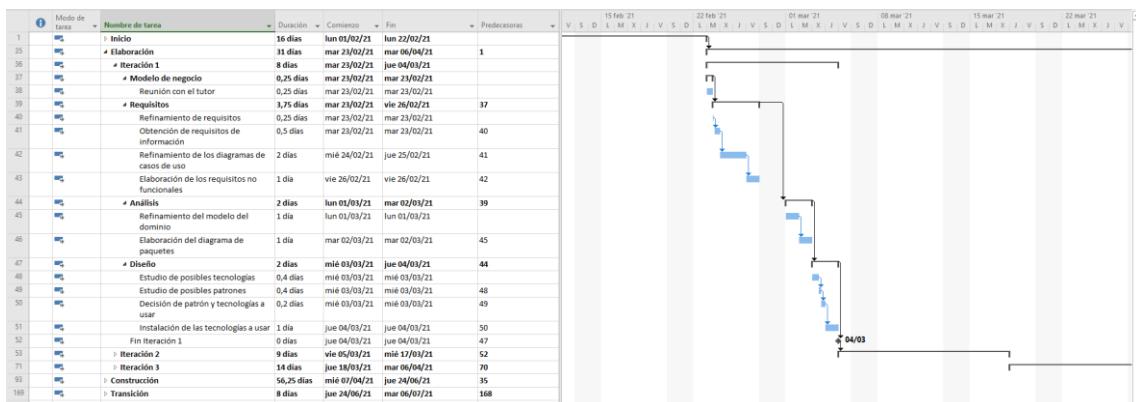


Ilustración 50: Elaboración, Iteración 1 Gnatt

	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	▷	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
35	▷	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	1
36	▷	▷ Iteración 1	8 días	mar 23/02/21	jue 04/03/21	
53	▷	▷ Iteración 2	9 días	vie 05/03/21	mié 17/03/21	52
54	▷	▷ Modelo de negocio	0,25 días	vie 05/03/21	vie 05/03/21	
55	▷	Reunión con el tutor	0,25 días	vie 05/03/21	vie 05/03/21	
56	▷	▷ Requisitos	1,75 días	vie 05/03/21	lun 08/03/21	54
57	▷	Refinamiento requisitos	0,5 días	vie 05/03/21	vie 05/03/21	
58	▷	Refinamiento diagramas de casos de uso	1 día	vie 05/03/21	lun 08/03/21	57
59	▷	Elaboración matriz rastreabilidad	0,25 días	lun 08/03/21	lun 08/03/21	58
60	▷	▷ Análisis	5 días	mar 09/03/21	lun 15/03/21	56
61	▷	Refinamiento del modelo de dominio	0,25 días	mar 09/03/21	mar 09/03/21	
62	▷	Refinamiento del diagrama de paquetes	0,75 días	mar 09/03/21	mar 09/03/21	61
63	▷	Elaboración de los diagramas de comunicación	1 día	mié 10/03/21	mié 10/03/21	62
64	▷	Elaboración de la vista de arquitectura	1 día	jue 11/03/21	jue 11/03/21	63
65	▷	Elaboración de los diagramas de secuencia	3 días	jue 11/03/21	lun 15/03/21	63
66	▷	▷ Diseño	1 día	mar 16/03/21	mar 16/03/21	60
67	▷	Elaboración de la primera versión de los diagramas de capas	1 día	mar 16/03/21	mar 16/03/21	
68	▷	▷ Desarrollo	1 día	mié 17/03/21	mié 17/03/21	66
69	▷	Inicio del desarrollo	1 día	mié 17/03/21	mié 17/03/21	
70	▷	Fin Iteración 2	0 días	mié 17/03/21	mié 17/03/21	68
71	▷	▷ Iteración 3	14 días	jue 18/03/21	mar 06/04/21	70
93	▷	▷ Construcción	56,25 días	mié 07/04/21	jue 24/06/21	35
169	▷	▷ Transición	8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 51: Elaboración, Iteración 2

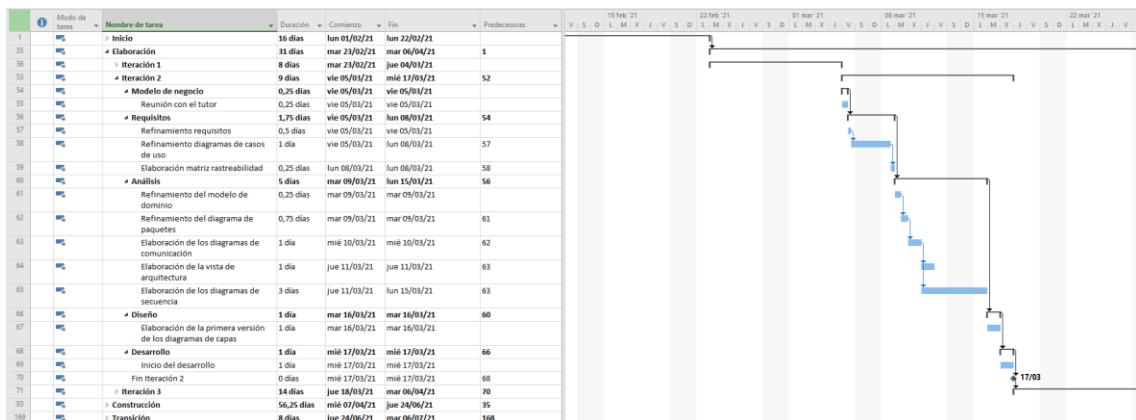


Ilustración 52: Elaboración, Iteración 2 Gnatt

	<b>i</b>	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		➡ Inicio		16 días	lun 01/02/21	lun 22/02/21	
35		➡ Elaboración		31 días	mar 23/02/21	mar 06/04/21	1
36		➡ Iteración 1		8 días	mar 23/02/21	jue 04/03/21	
53		➡ Iteración 2		9 días	vie 05/03/21	mié 17/03/21	52
71		➡ Iteración 3		14 días	jue 18/03/21	mar 06/04/21	70
72		➡ Modelo de negocio		0,25 días	jue 18/03/21	jue 18/03/21	
73		Reunión con el tutor		0,25 días	jue 18/03/21	jue 18/03/21	
74		➡ Requisitos		0,75 días	jue 18/03/21	jue 18/03/21	72
75		Refinar requisitos		0,25 días	jue 18/03/21	jue 18/03/21	
76		Refinar diagramas de casos de uso		0,5 días	jue 18/03/21	jue 18/03/21	75
77		➡ Análisis		2 días	vie 19/03/21	lun 22/03/21	74
78		Refinamiento del diagrama de paquetes		0,25 días	vie 19/03/21	vie 19/03/21	
79		Refinamiento de los diagramas de comunicación		0,5 días	vie 19/03/21	vie 19/03/21	78
80		Refinamiento de la vista de arquitectura		0,25 días	vie 19/03/21	vie 19/03/21	79
81		Refinamiento de los diagramas de secuencia		1 día	lun 22/03/21	lun 22/03/21	80
82		➡ Diseño		5 días	mar 23/03/21	lun 29/03/21	77
83		Refinamiento diagramas de capas		1 día	mar 23/03/21	mar 23/03/21	
84		Vista arquitectónica		1 día	mié 24/03/21	mié 24/03/21	83
85		Diagramas de secuencia diseño		3 días	jue 25/03/21	lun 29/03/21	84
86		➡ Desarrollo		5 días	mar 30/03/21	lun 05/04/21	82
87		Elaboración menú principal		2 días	mar 30/03/21	mié 31/03/21	
88		Elaboración listar viajeros		2 días	jue 01/04/21	vie 02/04/21	87
89		Elaboración añadir viajero		1 día	lun 05/04/21	lun 05/04/21	88
90		➡ Pruebas		1 día	mar 06/04/21	mar 06/04/21	86
91		Pruebas del código desarrollado		1 día	mar 06/04/21	mar 06/04/21	
92		Fin Iteración 3		0 días	mar 06/04/21	mar 06/04/21	90
93		➡ Construcción		56,25 días	mié 07/04/21	jue 24/06/21	35
169		➡ Transición		8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 53: Elaboración, Iteración 3

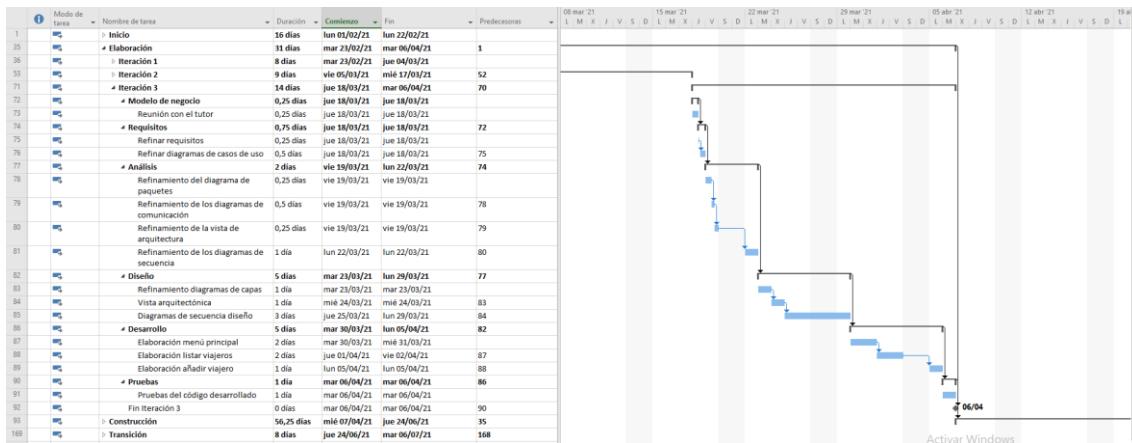


Ilustración 54: Elaboración, Iteración 3 Gnatt

### 5.2.3. Fase de construcción

En esta fase, se han continuado refinando el modelado de negocio, los requisitos y el análisis, se ha finalizado el diseño y se ha invertido una gran cantidad de tiempo en el desarrollo de la aplicación, incluidos sus algoritmos.

Esta fase finaliza con el desarrollo de una versión beta del producto.

	<i>i</i>	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		➡	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
35		➡	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	
93		➡	▷ Construcción	56,25 días	mié 07/04/21	jue 24/06/21	1
94		➡	▷ Iteración 1	10 días	mié 07/04/21	mar 20/04/21	
95		➡	▷ Modelado de negocio	0,5 días	mié 07/04/21	mié 07/04/21	
96		➡	Reunión con el tutor	0,5 días	mié 07/04/21	mié 07/04/21	
97		➡	▷ Requisitos	0,5 días	mié 07/04/21	mié 07/04/21	95
98		➡	Revisión y refinamiento	0,5 días	mié 07/04/21	mié 07/04/21	
99		➡	▷ Análisis	1 día	jue 08/04/21	jue 08/04/21	97
100		➡	Revisión y refinamiento	1 día	jue 08/04/21	jue 08/04/21	
101		➡	▷ Diseño	2 días	vie 09/04/21	lun 12/04/21	99
102		➡	Refinamiento de los diagramas de capas	0,5 días	vie 09/04/21	vie 09/04/21	
103		➡	Refinamiento vista arquitectónica	0,5 días	vie 09/04/21	vie 09/04/21	102
104		➡	Refinamiento de los diagramas de secuencia del diseño	1 día	lun 12/04/21	lun 12/04/21	103
105		➡	▷ Desarrollo	6 días	mar 13/04/21	mar 20/04/21	101
106		➡	Refinamiento Menú principal	0,5 días	mar 13/04/21	mar 13/04/21	
107		➡	Refinamiento listar viajeros	0,5 días	mar 13/04/21	mar 13/04/21	106
108		➡	Refinamiento añadir viajeros	1 día	mié 14/04/21	mié 14/04/21	107
109		➡	Elaboración modificar viajero	2 días	jue 15/04/21	vie 16/04/21	108
110		➡	Elaboración eliminar viajero	2 días	lun 19/04/21	mar 20/04/21	109
111		➡	▷ Pruebas	0 días	mar 20/04/21	mar 20/04/21	105
112		➡	Pruebas de la gestión de viajeros	0 días	mar 20/04/21	mar 20/04/21	
113		➡	Fin de Iteración 1	0 días	mar 20/04/21	mar 20/04/21	111
114		➡	▷ Iteración 2	26 días	mié 21/04/21	mié 26/05/21	113
132		➡	▷ Iteración 3	14 días	jue 27/05/21	mar 15/06/21	131
153		➡	▷ Iteración 4	6,25 días	mié 16/06/21	jue 24/06/21	152
169		➡	▷ Transición	8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 55: Construcción, Iteración 1

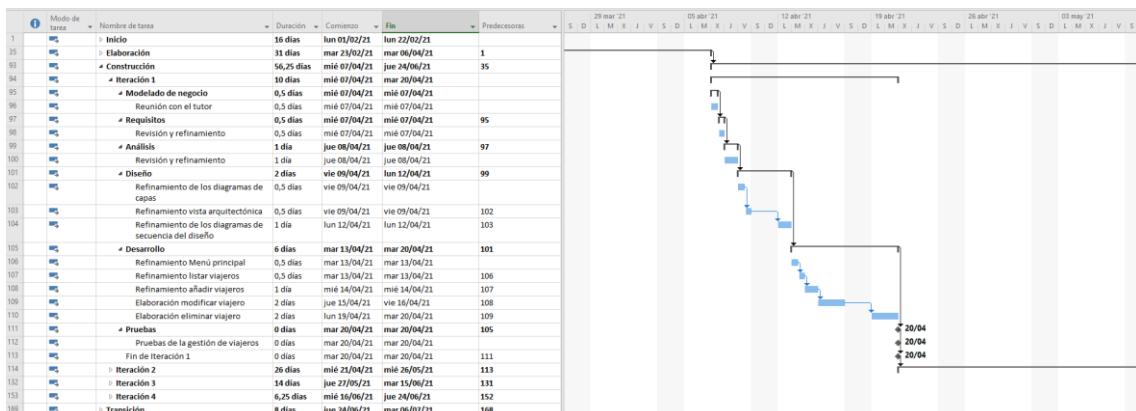


Ilustración 56: Construcción, Iteración 1 Gnatt

	<b>Modo de tarea</b>	<b>Nombre de tarea</b>	<b>Duración</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Predecesoras</b>
1	▷	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
35	▷	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	1
93	▷	▲ Construcción	56,25 días	mié 07/04/21	jue 24/06/21	35
94	▷	▷ Iteración 1	10 días	mié 07/04/21	mar 20/04/21	
114	▷	▲ Iteración 2	26 días	mié 21/04/21	mié 26/05/21	113
115	▷	▲ Modelado de negocio	0,5 días	mié 21/04/21	mié 21/04/21	
116	▷	Reunión con el tutor	0,5 días	mié 21/04/21	mié 21/04/21	
117	▷	▲ Requisitos	0,25 días	mié 21/04/21	mié 21/04/21	115
118	▷	Revisión y refinamiento	0,25 días	mié 21/04/21	mié 21/04/21	
119	▷	▲ Análisis	0,25 días	mié 21/04/21	mié 21/04/21	117
120	▷	Revisión y refinamiento	0,25 días	mié 21/04/21	mié 21/04/21	
121	▷	▲ Diseño	1 día	jue 22/04/21	jue 22/04/21	119
122	▷	Revisión y refinamiento	1 día	jue 22/04/21	jue 22/04/21	
123	▷	▲ Desarrollo	21 días	vie 23/04/21	vie 21/05/21	121
124	▷	Refinamiento de la gestión de viajeros	2 días	vie 23/04/21	lun 26/04/21	
125	▷	Desarrollo de la pantalla de rueda	2 días	mar 27/04/21	mié 28/04/21	124
126	▷	Desarrollo del algoritmo de generación de horarios	15 días	jue 29/04/21	mié 19/05/21	125
127	▷	Desarrollo del algoritmo de ponderación de horarios	2 días	jue 20/05/21	vie 21/05/21	126
128	▷	▲ Pruebas	3 días	lun 24/05/21	mié 26/05/21	123
129	▷	Pruebas de gestión de viajeros	0,5 días	lun 24/05/21	lun 24/05/21	
130	▷	Pruebas de los algoritmos desarrollados	2,5 días	lun 24/05/21	mié 26/05/21	129
131	▷	Fin iteración 2	0 días	mié 26/05/21	mié 26/05/21	128
132	▷	▷ Iteración 3	14 días	jue 27/05/21	mar 15/06/21	131
153	▷	▷ Iteración 4	6,25 días	mié 16/06/21	jue 24/06/21	152
169	▷	▷ Transición	8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 57: Construcción, Iteración 2

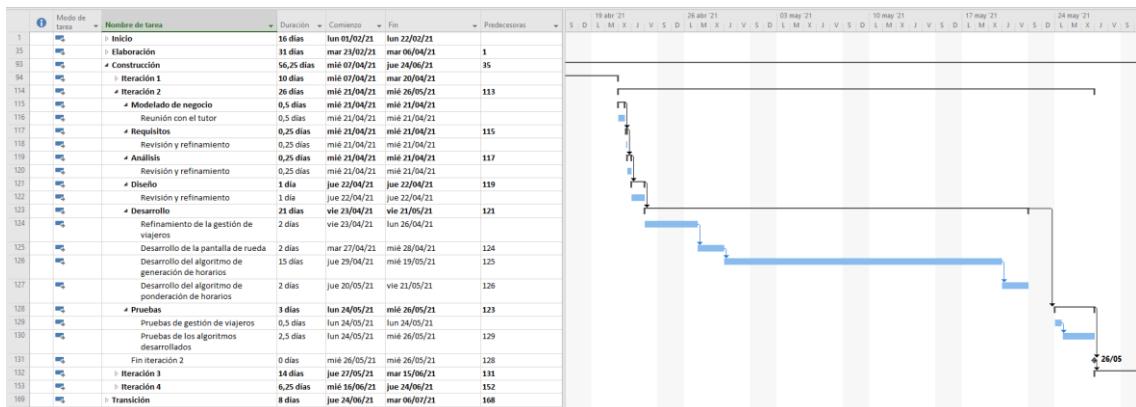


Ilustración 58: Construcción, Iteración 2 Gnatt

	<b>i</b>	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		▷ Inicio		16 días	lun 01/02/21	lun 22/02/21	
35		▷ Elaboración		31 días	mar 23/02/21	mar 06/04/21	1
93		△ Construcción		56,25 días	mié 07/04/21	jue 24/06/21	35
94		▷ Iteración 1		10 días	mié 07/04/21	mar 20/04/21	
114		▷ Iteración 2		26 días	mié 21/04/21	mié 26/05/21	113
132		△ Iteración 3		14 días	jue 27/05/21	mar 15/06/21	131
133		△ Modelo de negocio		0,5 días	jue 27/05/21	jue 27/05/21	
134		Reunión con el tutor		0,5 días	jue 27/05/21	jue 27/05/21	
135		△ Requisitos		0,25 días	jue 27/05/21	jue 27/05/21	133
136		Revisión y refinamiento		0,25 días	jue 27/05/21	jue 27/05/21	
137		△ Análisis		0,25 días	jue 27/05/21	jue 27/05/21	135
138		Revisión y refinamiento		0,25 días	jue 27/05/21	jue 27/05/21	
139		△ Diseño		0,5 días	vie 28/05/21	vie 28/05/21	137
140		Revisión y refinamiento		0,5 días	vie 28/05/21	vie 28/05/21	
141		△ Desarrollo		10,5 días	vie 28/05/21	vie 11/06/21	139
142		Refinamiento de la gestión de viajeros		0,5 días	vie 28/05/21	vie 28/05/21	
143		Refinamiento de la pantalla de rueda		1 día	lun 31/05/21	lun 31/05/21	142
144		Refinamiento del algoritmo de generación de horarios		3 días	mar 01/06/21	jue 03/06/21	143
145		Refinamiento del algoritmo de ponderación de horarios		1 día	vie 04/06/21	vie 04/06/21	144
146		Desarrollo de la generación de PDF		3 días	lun 07/06/21	mié 09/06/21	145
147		Desarrollo de la gestión de reinicio		2 días	jue 10/06/21	vie 11/06/21	146
148		△ Pruebas		2 días	lun 14/06/21	mar 15/06/21	141
149		Pruebas gestión de viajeros		0,5 días	lun 14/06/21	lun 14/06/21	
150		Pruebas gestión de rueda		1 día	lun 14/06/21	mar 15/06/21	149
151		Pruebas gestión de reinicio		0,5 días	mar 15/06/21	mar 15/06/21	150
152		Fin iteración 3		0 días	mar 15/06/21	mar 15/06/21	148
153		▷ Iteración 4		6,25 días	mié 16/06/21	jue 24/06/21	152
169		▷ Transición		8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 59: Construcción, Iteración 3

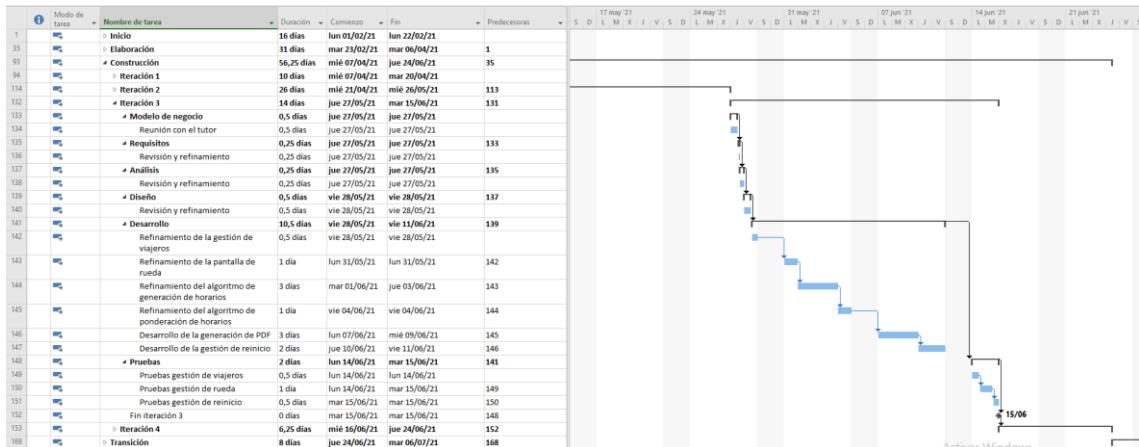


Ilustración 60: Construcción, Iteración 3 Gnatt

	<b>Modo de tarea</b>	Nombre de tarea	<b>Duración</b>	Comienzo	Fin	Predecesoras
1	➡	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
35	➡	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	1
93	➡	▫ Construcción	56,25 días	mié 07/04/21	jue 24/06/21	35
94	➡	▷ Iteración 1	10 días	mié 07/04/21	mar 20/04/21	
114	➡	▷ Iteración 2	26 días	mié 21/04/21	mié 26/05/21	113
132	➡	▷ Iteración 3	14 días	jue 27/05/21	mar 15/06/21	131
153	➡	▫ Iteración 4	6,25 días	mié 16/06/21	jue 24/06/21	152
154	➡	▫ Modulo de negocio	0,5 días	mié 16/06/21	mié 16/06/21	
155	➡	Reunión con el tutor	0,5 días	mié 16/06/21	mié 16/06/21	
156	➡	▫ Requisitos	0,25 días	mié 16/06/21	mié 16/06/21	154
157	➡	Revisión y refinamiento	0,25 días	mié 16/06/21	mié 16/06/21	
158	➡	▫ Análisis	0,25 días	mié 16/06/21	mié 16/06/21	156
159	➡	Revisión y refinamiento	0,25 días	mié 16/06/21	mié 16/06/21	
160	➡	▫ Diseño	0,25 días	jue 17/06/21	jue 17/06/21	158
161	➡	Revisión y refinamiento	0,25 días	jue 17/06/21	jue 17/06/21	
162	➡	▫ Desarrollo	3 días	jue 17/06/21	mar 22/06/21	160
163	➡	Refinamiento de la gestión de viajeros	0,5 días	jue 17/06/21	jue 17/06/21	
164	➡	Refinamiento de la gestión de rueda	2 días	jue 17/06/21	lun 21/06/21	163
165	➡	Refinamiento de la gestión de reinicio	0,5 días	lun 21/06/21	mar 22/06/21	164
166	➡	▫ Pruebas	2 días	mar 22/06/21	jue 24/06/21	162
167	➡	Versión beta	2 días	mar 22/06/21	jue 24/06/21	
168	➡	Fin de la Iteración 4	0 días	jue 24/06/21	jue 24/06/21	166
169	➡	▷ Transición	8 días	jue 24/06/21	mar 06/07/21	168

Ilustración 61: Construcción, Iteración 4

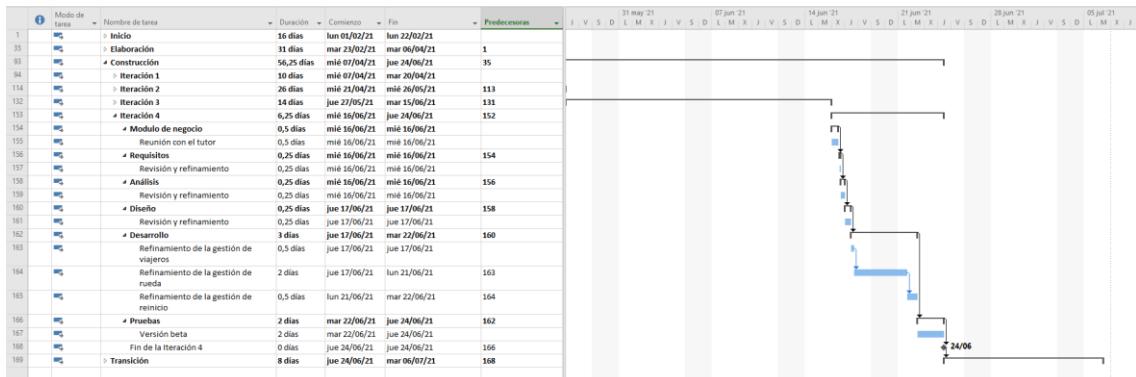


Ilustración 62: Construcción, Iteración 4 Gnatt

### 5.2.4. Fase de transición

En la transición se ha culminado el trabajo y se ha obtenido la versión de la aplicación que se mostrará en la defensa de esta.

	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	▶	▷ Inicio	16 días	lun 01/02/21	lun 22/02/21	
35	▶	▷ Elaboración	31 días	mar 23/02/21	mar 06/04/21	1
93	▶	▷ Construcción	56,25 días	mie 07/04/21	jue 24/06/21	35
169	▶	△ Transición	8 días	jue 24/06/21	mar 06/07/21	168
170	▶	△ Iteración 1	8 días	jue 24/06/21	mar 06/07/21	
171	▶	△ Modelo de negocio	1 día	jue 24/06/21	vie 25/06/21	
172	▶	Reunión con el tutor	1 día	jue 24/06/21	vie 25/06/21	
173	▶	△ Implementación	5 días	vie 25/06/21	vie 02/07/21	171
174	▶	Últimas mejoras	2 días	vie 25/06/21	mar 29/06/21	
175	▶	Obtención de la versión final	3 días	mar 29/06/21	vie 02/07/21	174
176	▶	△ Pruebas	2 días	vie 02/07/21	mar 06/07/21	173
177	▶	Pruebas finales	2 días	vie 02/07/21	mar 06/07/21	
178	▶	Fin Iteración 1	0 días	mar 06/07/21	mar 06/07/21	176

Ilustración 63: Transición, Iteración 1

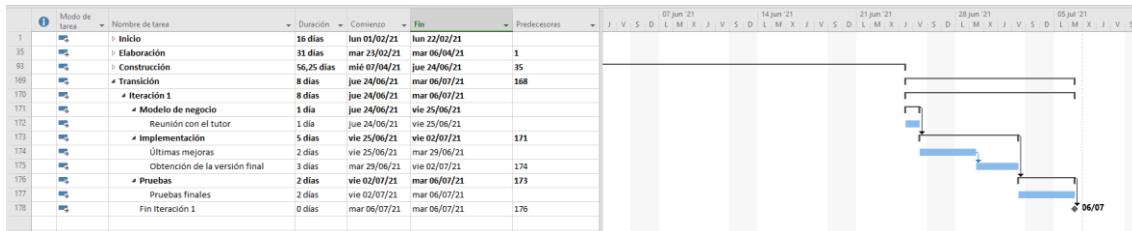


Ilustración 64: Transición, Iteración 1 Gnatt

## 5.3. Especificación de requisitos

La Especificación de Requisitos del Sistema (ERS) ha de evaluar las necesidades que quieren satisfacer los clientes mediante el producto que desean desarrollar, así como las de los usuarios que disfrutarán del mismo.

Por tanto, es necesario que la ERS desarrollada sirva a su vez como el medio de comunicación entendible por todos los participantes que tomarán parte en el desarrollo y explotación del sistema (clientes, usuarios, ingenieros de requisitos y desarrolladores entre otros). (Andalucía, s.f.)

En este apartado se presentarán a todos los participantes del proyecto (individuos y organizaciones), se explicará el método mediante el cual se obtuvieron los requisitos, se definirá a los actores del sistema, los requisitos funcionales y no funcionales, se especificarán los casos de uso del sistema y finalmente se correlacionarán todos los elementos explicados mediante la matriz de rastreabilidad. Esta información estará ampliamente recogida en el Anexo 2.

### 5.3.1. Objetivos

Los objetivos que se persiguen al realizar esta aplicación son los siguientes:

- Elaboración de un sistema que permita optimizar los viajes por trabajo
- Gestión de viajeros
- Gestión de rueda
- Reinicio de aplicación

Tabla 1: OBJ-003 Gestión de rueda

OBJ-003		Gestión de Rueda
Versión	1.0 (19/08/2021)	

Autores	Marcos Unzueta Puente
Descripción	<p>La aplicación llama “rueda” al horario de desplazamiento al trabajo óptimo, generado mediante los algoritmos desarrollados a partir de los datos de viajeros brindados por el usuario al sistema.</p> <p>El sistema, en primer lugar, ha de ser capaz de generar el horario de desplazamientos al centro óptimo haciendo uso de los datos de viajeros.</p> <p>El sistema ha de ser capaz de desarrollar un fichero con el horario generado, de tal manera que la “rueda” calculada se pueda exponer a los interesados de una manera visual y entendible.</p> <p>El sistema ha de permitir compartir el horario óptimo generado a partir de otras aplicaciones instaladas en el dispositivo móvil (WhatsApp, Gmail, Telegram, etc.).</p> <p>Ha de implementarse un método sencillo y rápido de actualizar el horario cuando haya pequeños cambios.</p>
Subobjetivos	
Importancia	Alta
Urgencia	Alta
Estado	Completado
Comentarios	

### 5.3.2. Requisitos de información

Son los requisitos que indican la información con la que debe trabajar el software y que permitirán cumplir con éxito los objetivos que se han marcado previamente en la elaboración de este informe. (juntadeandalucia, s.f.)

- Información de viajeros
- Información de Horario de Trabajo Viajero
- Información del Día de Horario de Trabajo Viajero
- Información del Horario de Rueda
- Dia de Coches
- Dia de Coche

Tabla 2: IRQ-004 Información del Horario de Rueda

IRQ-004	Información del Horario de Rueda
Versión	1.0 (19/08/2021)
Autores	Marcos Unzueta Puente
Descripción	<p>Es sistema ha de ser capaz de generar, almacenar y eliminar los datos del horario óptimo de rueda para los viajeros dados.</p> <p>Será necesario almacenar una lista con cada uno de los días de la semana (día de coches), que contenga a su vez los coches que se desplazan ese día.</p> <p>Es necesario conocer también los datos que posteriormente se usarán para considerar cuál de los horarios generados es el óptimo,</p>

	los minutos esperados de manera innecesaria por los viajeros en el horario generado y el número total de coches que se lleva. Estos datos se usarán para calcular la bondad del horario mediante un algoritmo. La bondad indicará que horario es mejor.
Requisitos asociados	OBJ-001,OBJ-003
Datos específicos	- Días de coches - Minutos esperados - Total de coches llevados - Bondad
Estado	Completado
Importancia	Alta
Urgencia	Alta
Comentarios	

### 5.3.3. Requisitos funcionales

La guía Business Analysis Body of Knowledge (BAOBK) en la versión 3 contiene la siguiente definición para los requisitos funcionales: "Son las descripciones explícitas del comportamiento que debe tener una solución de software y que información debe manejar." ((BABOK), s.f.)

Un diagrama de casos de uso se utiliza para modelar los requisitos funcionales del sistema, mostrando así la funcionalidad de este. (Sevilla, s.f.)

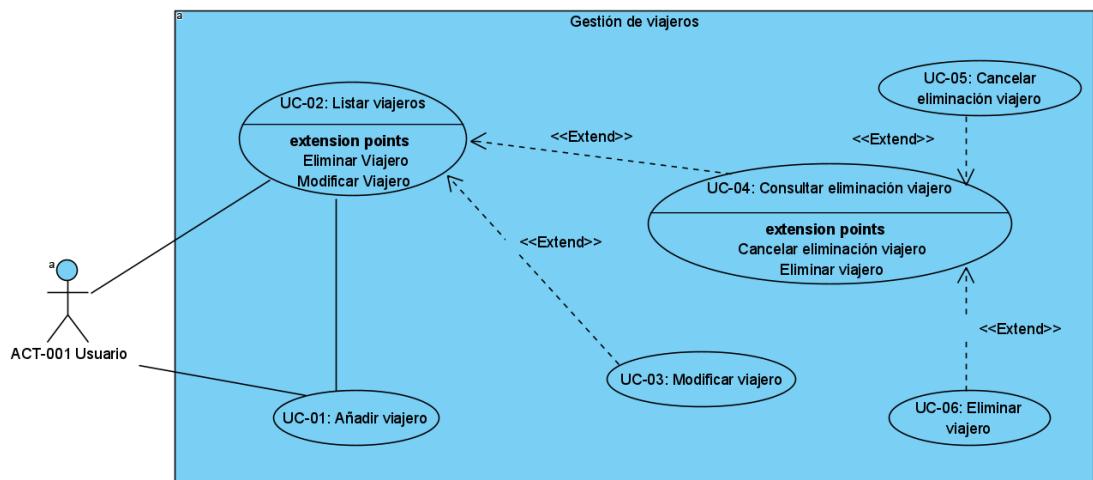


Ilustración 65: Diagrama de casos de uso de gestión de viajeros

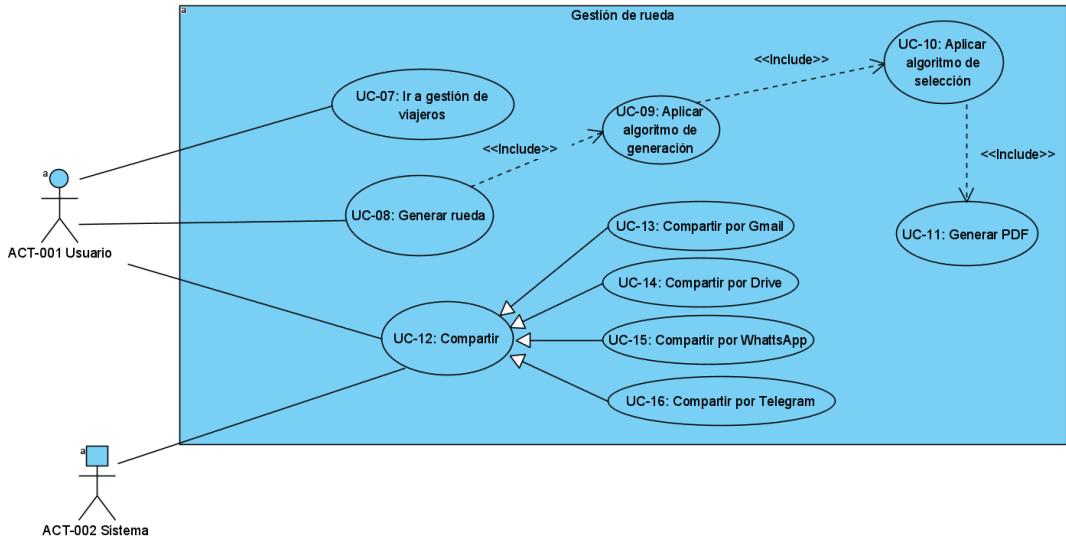


Ilustración 66: Diagrama de casos de uso de gestión de rueda

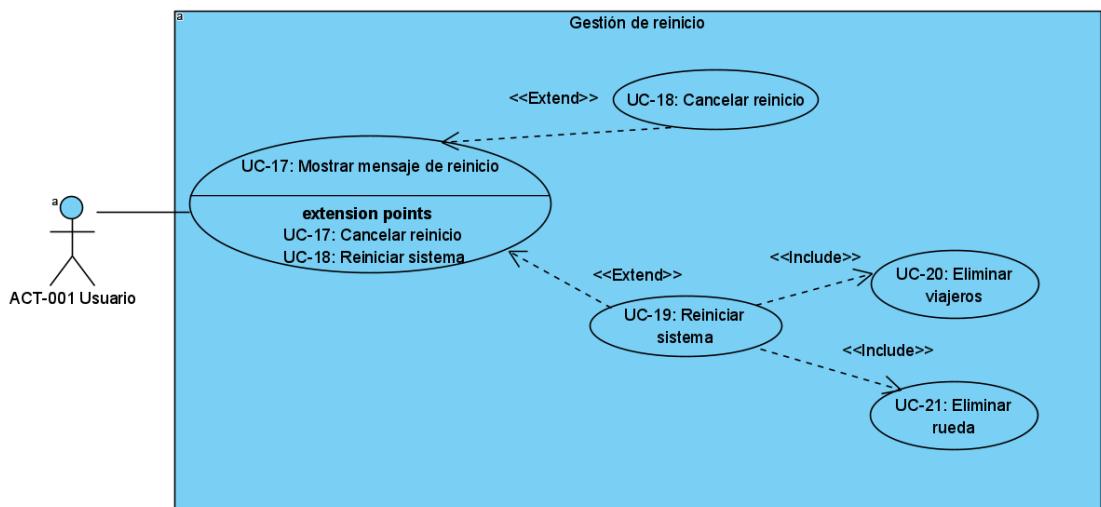


Ilustración 67: Diagrama de casos de uso de gestión de reinicio

Tabla 3: UC-001 Añadir viajero

<b>UC-001</b>		<b>Añadir viajero</b>
Versión	1.0 (19/08/2021)	
Autores	Marcos Unzueta Puente	
Objetivos asociados	OBJ-001, OBJ-002	
Requisitos asociados		
Descripción	El actor ACT-001 Usuario añade un viajero.	
Actores	ACT-001 Usuario	

Precondición	Haber inicializado la aplicación y haber accedido a la sección de viajeros.
Secuencia normal	<ol style="list-style-type: none"> <li>1. El caso de uso se inicia cuando el actor ACT-001 Usuario selecciona “Añadir Viajero”.</li> <li>2. La aplicación muestra un formulario de registro del viajero.</li> <li>3. El actor ACT-001 Usuario rellena el formulario.</li> <li>4. El actor ACT-001 pulsa el botón “Añadir”.</li> <li>5. La aplicación comprueba que los parámetros introducidos son correctos.</li> <li>6. La aplicación registra al viajero en la base de datos.</li> <li>7. La aplicación actualiza la lista de viajeros que se muestra al usuario.</li> </ol>
Postcondición	
Excepciones	<p>En el paso 4: Si el sistema detecta que el actor ACT-001 Usuario ha introducido valores erróneos en el formulario, como un nombre o un apellido muy largo, no ha marcado ningún día como laboral o ha marcado alguna salida a una hora anterior a la entrada.</p> <p>Si no se soluciona el error este caso de uso queda sin efecto.</p> <p>En el paso 6: Si el sistema tiene problemas a la hora de almacenar el viajero en la base de datos se indica al ACT-001 Usuario y el caso de uso queda sin efecto.</p>
Comentarios	

### 5.3.4. Requisitos no funcionales:

En este apartado se detallarán los requisitos no funcionales del sistema a desarrollar.

Los requisitos no funcionales son las condiciones que impone el cliente, pero que no se tratan información a guardar ni funciones a realizar.

Estos requisitos se corresponden, generalmente, con características de funcionamiento que van asociados, principalmente, con aspectos referentes a la calidad. (Cuajimalpa, s.f.)

- Funcionamiento de la aplicación en el S.O Android
- Interfaz amigable y sencilla
- Adaptabilidad y escalabilidad
- Eficacia

Tabla 4: NRF-001 Funcionamiento de la aplicación en el S.O Android

NFR-001	Funcionamiento de la aplicación en el S.O Android
Versión	1.0 (19/08/2021)
Autores	Marcos Unzueta Puente
Descripción	<p>El software ha de poder instalarse y funcionar en el sistema operativo Android.</p> <p>Android es el sistema operativo más usado para móviles y, por tanto, el que hará más accesible la aplicación para los usuarios interesados.</p>

Dependencias	
Importancia	Alta
Comentarios	

## 5.4. Análisis del sistema

Una vez se hayan definido correctamente los requisitos y los objetivos del software, se procede a realizar el análisis del sistema.

Se tratará de proporcionar una visión realista del modelo de dominio, se mostrarán los paquetes del sistema y se generarán diagramas de secuencia de los casos de uso.

El análisis del sistema se encuentra ampliamente detallado en el anexo 3.

### 5.4.1. Modelo de dominio

El modelo de dominio sirve para representar, de manera visual, las clases del mundo real con las que se generará el sistema, así como las relaciones existentes entre ellas y los atributos con los que cuentan.

El modelo de dominio se elabora a partir de diagramas de clases en los que no se definen las operaciones. (exactas, s.f.)

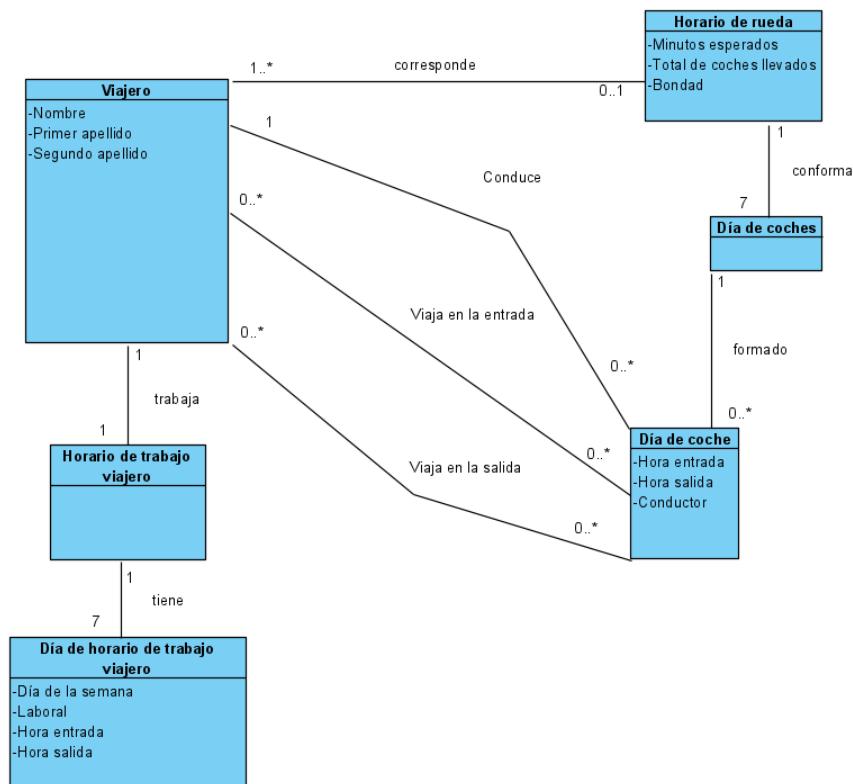


Ilustración 68: Modelo de dominio

#### 5.4.2. Vista de arquitectura del modelo de análisis

Esta vista de arquitectura pretende ser una abstracción del sistema desde una perspectiva seleccionada específicamente para mostrar al lector una visión global del software a desarrollar. (rploaiza, s.f.)

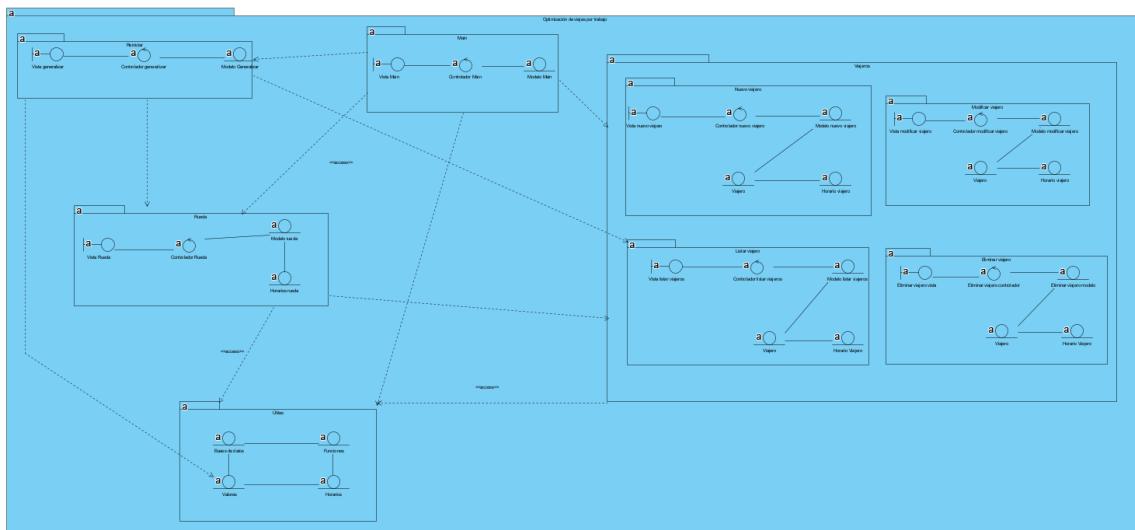


Ilustración 69: Vista arquitectónica

Como la vista arquitectónica es muy grande y no permite una buena lectura, se ha descompuesto la imagen en tres para mejorar su visibilidad.

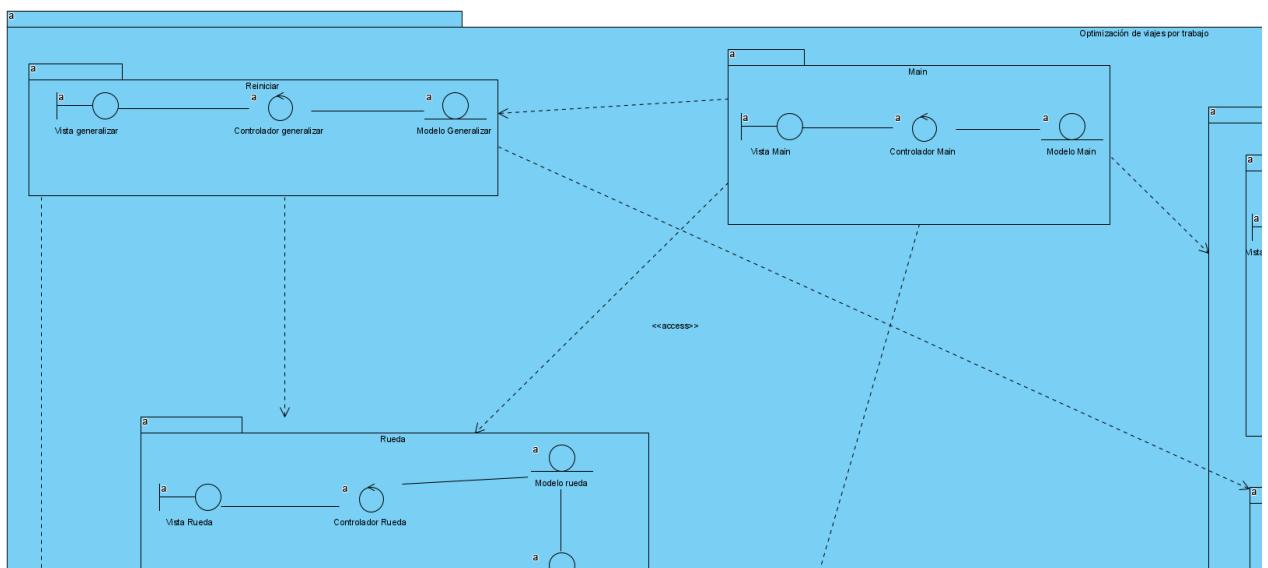


Ilustración 70: Vista arquitectónica arriba izquierda

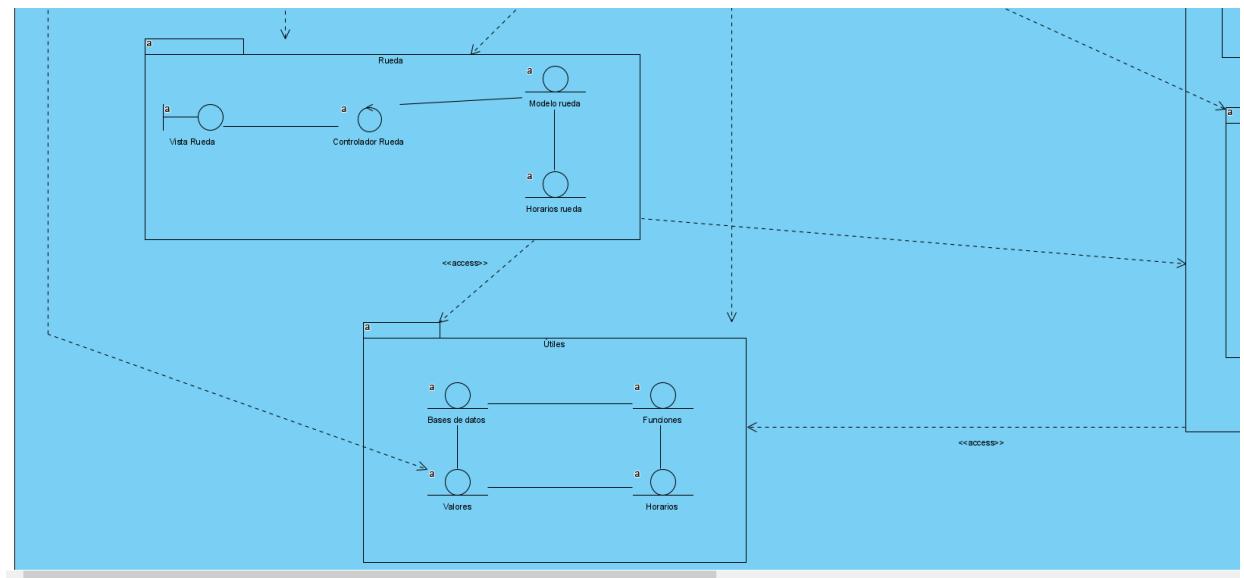


Ilustración 71: Vista arquitectónica abajo izquierda

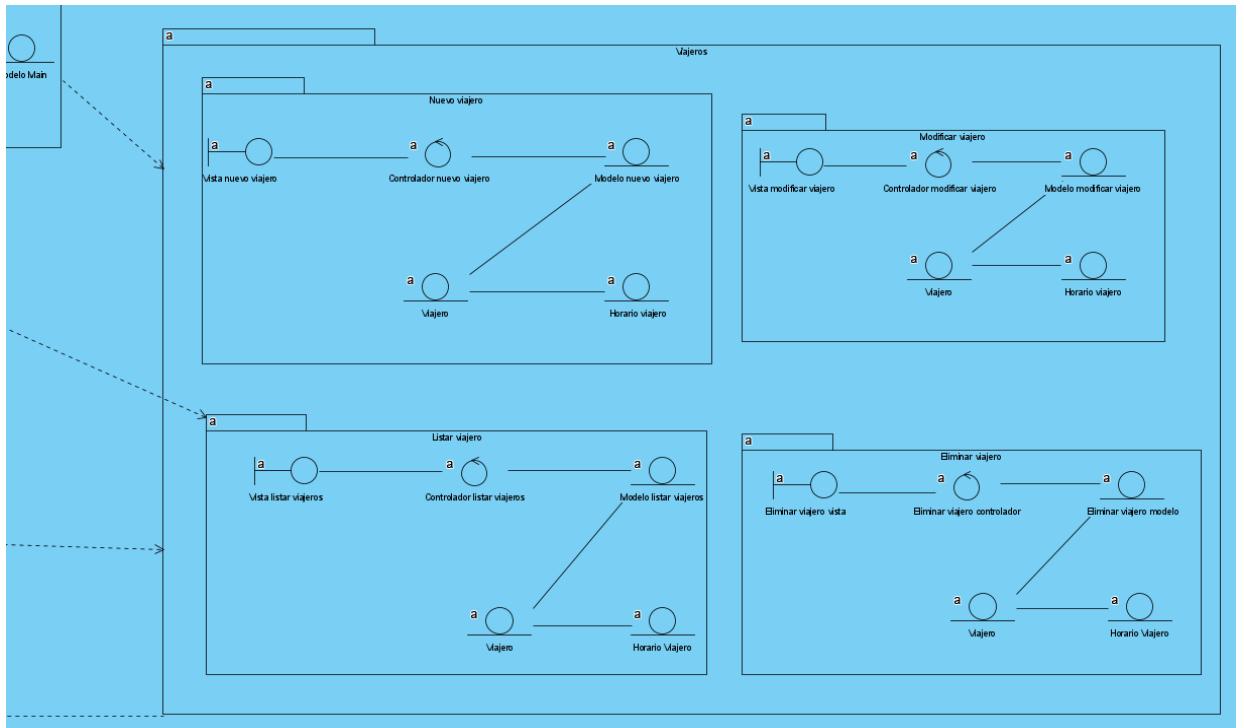


Ilustración 72: Vista arquitectónica derecha

#### 5.4.3. Realización de casos de uso del modelo de análisis

Se procede a describir y representar las interacciones y mensajes enviados entre los elementos de cada caso de uso, ya definidos en el anexo anterior, haciendo uso de los diagramas de secuencia.

Los diagramas de secuencia forman parte de los diagramas de interacción.

Se encargan de mostrar la propia interacción entre los objetos del caso de uso poniendo especial énfasis en el intercambio de mensajes entre estos.

Se refinarán posteriormente, pero, esta primera aproximación es muy útil para elaborar una idea inicial de cómo elaborar los distintos casos de uso del proyecto.

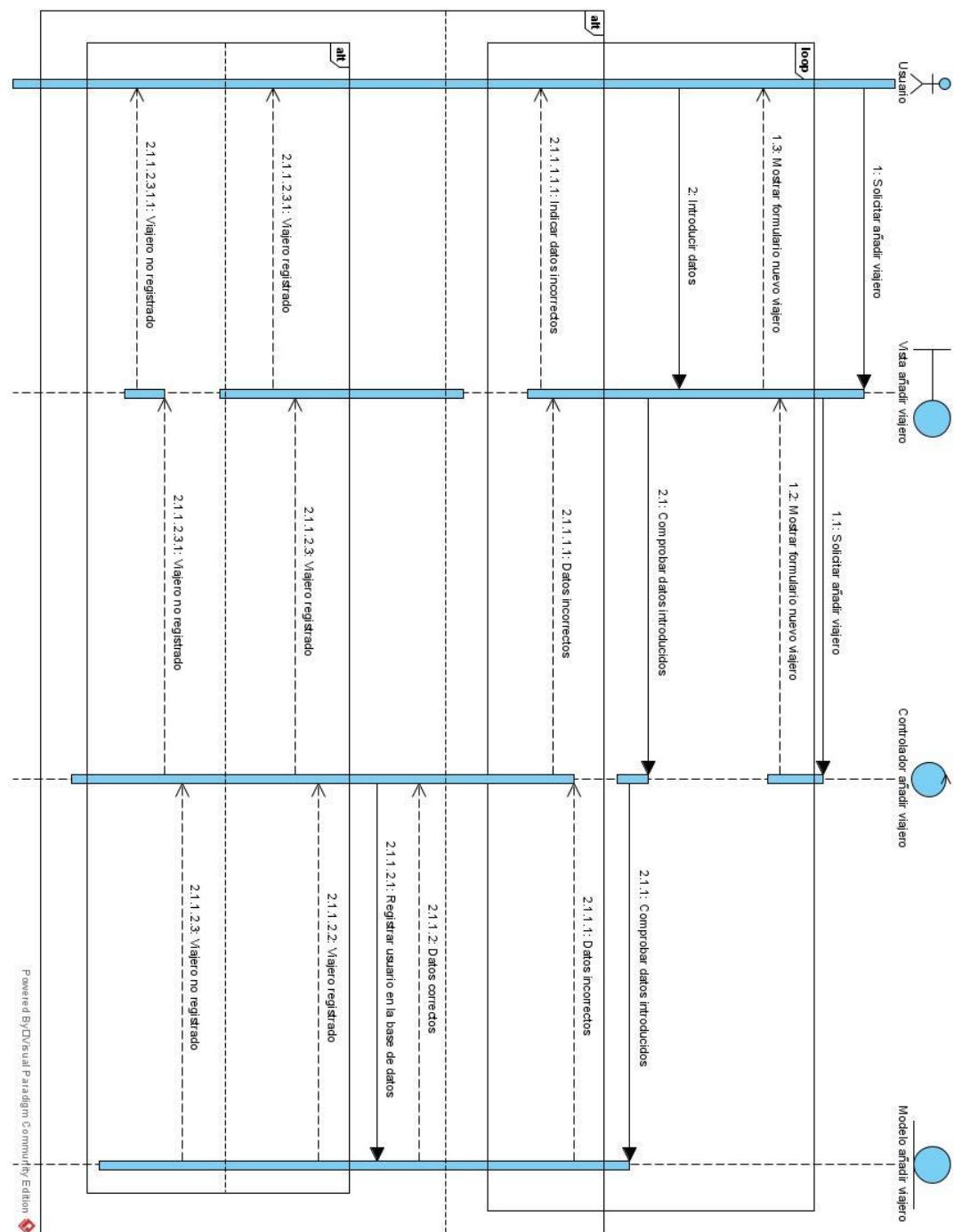


Ilustración 73: Diagrama de secuencia añadir viajero

## 5.5. Diseño del sistema

En el diseño del sistema se tratará de seguir refinando el proyecto en el que se está trabajando, siendo este apartado el último que trata enteramente de ingeniería del software.

Se utilizará la información obtenida en el análisis del sistema llevado a cabo para elaborar el dominio de la solución.

Se profundizará sobre el patrón arquitectónico escogido, se realizarán diagramas de las clases de diseño teniéndolo en cuenta, se realizará un diagrama que muestre la vista de arquitectura, se refinará los diagramas de secuencia realizados en el anexo anterior llevándolos a un nivel de abstracción más bajo y se realizará un diagrama de la vista de despliegue.

Toda la información sobre el diseño del sistema se encuentra recogida en el anexo 4.

### 5.5.1. Patrón arquitectónico

Tras un análisis exhaustivo, se decidió utilizar el patrón Modelo-Vista-Controlador por una serie de motivos que se explicarán posteriormente.

Es importante tener en cuenta que, debido a la popularidad de este patrón, existen algunas variantes. En este documento se va a considerar la más común.

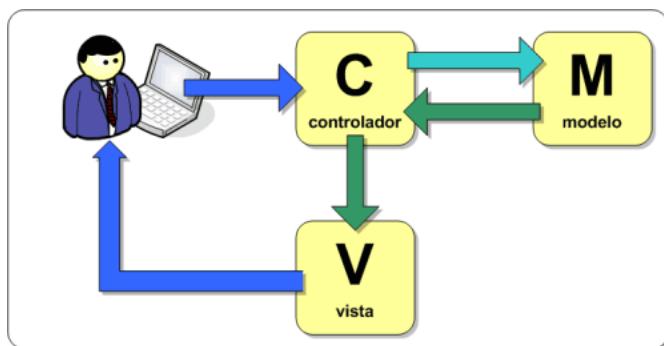


Ilustración 74: Patrón Modelo-Vista-Controlador

El principio básico de este paradigma, que ayuda a organizar y estructurar los componentes del sistema software a desarrollar, consiste en separar los componentes de una aplicación en tres capas principales (campusmvp, s.f.):

- **Vista:**

La vista se encarga de generar la interfaz con la que se comunicará el usuario.

En el caso concreto de este proyecto, la vista está formada por los archivos con la extensión xml que se encuentran en el directorio res.layout del proyecto y forman la interfaz de la aplicación.

Al programar los ficheros xml, solo se desarrolla la parte visual, en ningún momento se profundiza en las funcionalidades de la aplicación.

La vista ha de contener los elementos que permitan al usuario interactuar y enviar información, de tal manera que este pueda solicitar la realización de acciones del sistema.

- **Controlador:**

La función del controlador es actuar como intermediario entre el propio usuario y el sistema. Ha de ser capaz de reconocer las interacciones sobre la vista de la aplicación, como la pulsación de un botón o el deslizamiento en un slidebar. (material.io, s.f.)

Cuando el usuario dispare acciones del sistema interactuando con elementos de la vista, el controlador será el encargado de coordinar la respuesta, que puede ser desde cargar una nueva vista hasta solicitar al modelo que realice acciones.

En el caso de la aplicación desarrollada, es el encargado de reconocer los elementos XML de la vista, así como las acciones que el usuario lleva a cabo sobre ellos, respondiendo tanto en el tratamiento de datos (modelo) como en la información que se muestra (vista).

Se ha elegido utilizar Java como lenguaje para los controladores de la aplicación. Estos se encontrarán en subdirectorios de la carpeta java.com.marcosunzueta.larueda y están indicados por la palabra controlador.

Otra misión importante que tienen es la de ejercer de traductor entre modelo y vista, interpretando los datos que se obtienen la vista y traduciéndolos para el modelo. También se realiza el proceso inverso.

- **Modelo:**

En el modelo se encuentran los datos que maneja el sistema, así como la lógica del negocio.

En el caso de esta aplicación Android desarrollada en el entorno de desarrollo integrado Android Studio, se utilizan ficheros programados con el lenguaje java, que almacenarán la información en atributos y operarán haciendo uso de métodos creados.

Una función importante del modelo de esta aplicación es la de gestionar la interacción con la base de datos que almacenará la información. (androidauthority, s.f.)

### 5.5.2. Capas del diseño

Se mostrarán los paquetes que posee cada una de las capas generadas por la aplicación del patrón Modelo-Vista-Controlador.

En cada paquete se podrán ver las clases y archivos que contiene.

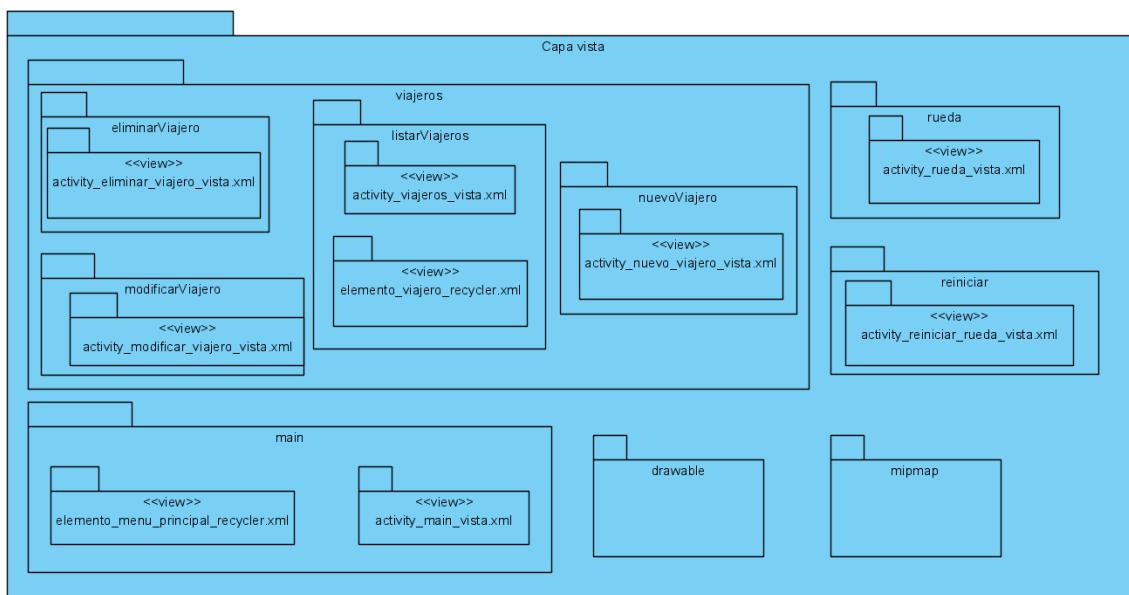


Ilustración 75: Diagrama capa vista

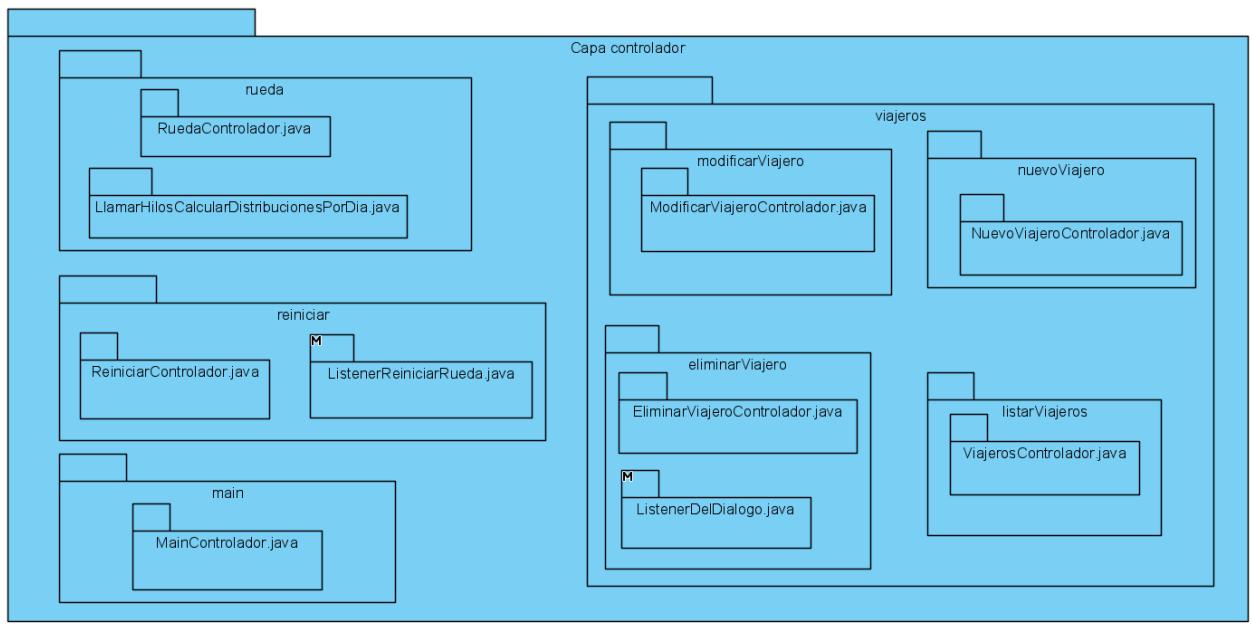


Ilustración 76: Diagrama capa controlador

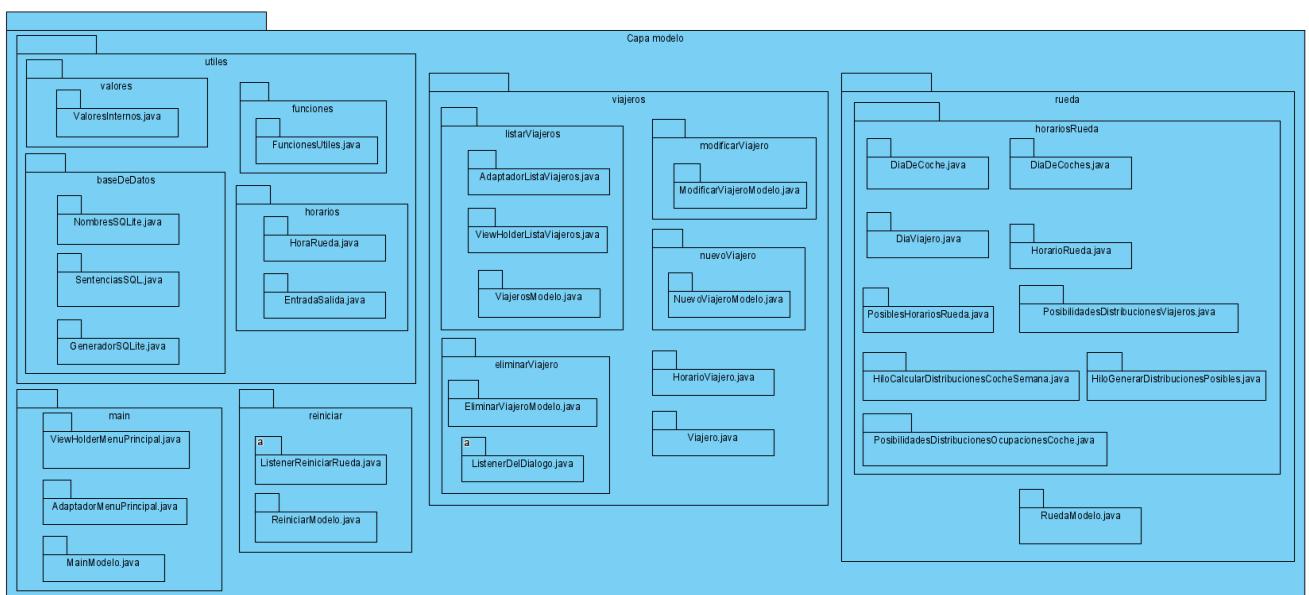


Ilustración 77: Diagrama capa modelo

### 5.5.3. Vista arquitectónica

Se procede a mostrar una vista arquitectónica global, marcada por el patrón Modelo-Vista-Controlador utilizado y en la que se hace referencia a las capas ya explicadas en el punto anterior.

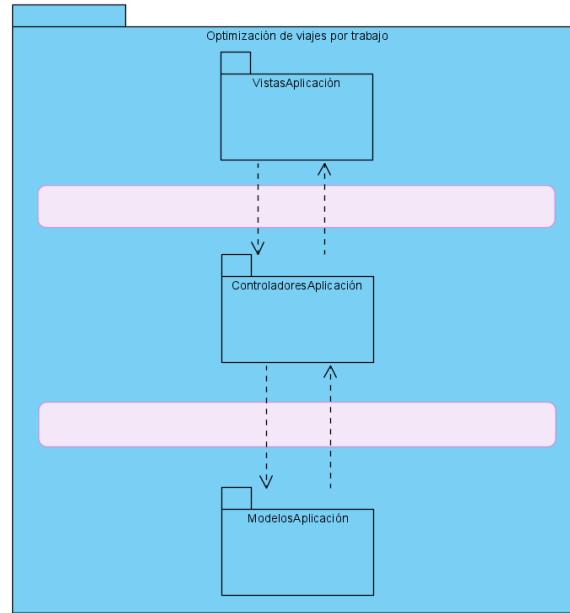


Ilustración 78: Vista arquitectónica

#### 5.5.4. Modelo de despliegue

El objetivo de esta sección es representar gráficamente los elementos físicos que componen el sistema y la asignación de los componentes software en estos elementos. (manuel.cillero, s.f.)

Representa tanto la arquitectura software como la hardware.

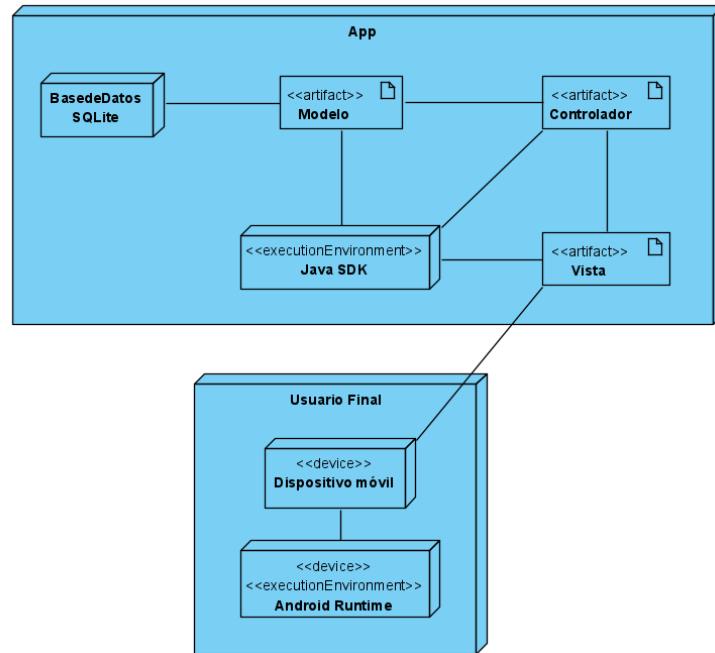


Ilustración 79: Modelo de despliegue

#### 5.5.5. Realización de casos de uso del modelo de diseño

Se refinarán los diagramas de secuencia realizados previamente.

A continuación, se mostrará el mismo diagrama de secuencia refinado.

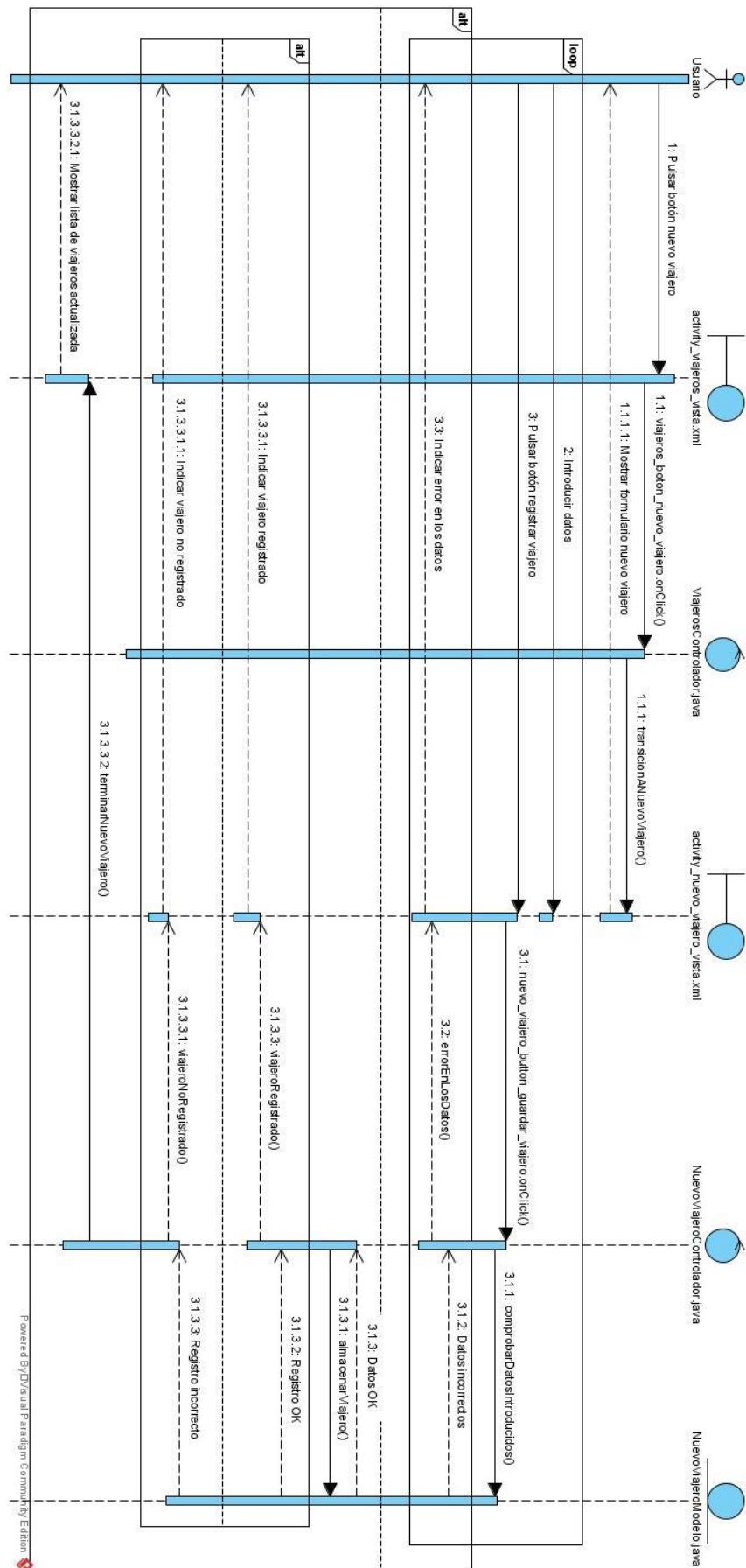


Ilustración 80: Diagrama de secuencia añadir viajero diseño

## 5.6. Documentación técnica sobre la aplicación

El producto software generado para la realización de este Trabajo de Fin de Máster se ha programado haciendo uso del entorno de desarrollo integrado Android Studio, por tanto, la distribución de directorios y ficheros estará condicionada por la estructura por defecto existente en el propio entorno mencionado.

Esta información se encuentra recogida en el anexo 5.

El espacio de trabajo utilizado tendrá la siguiente estructura:

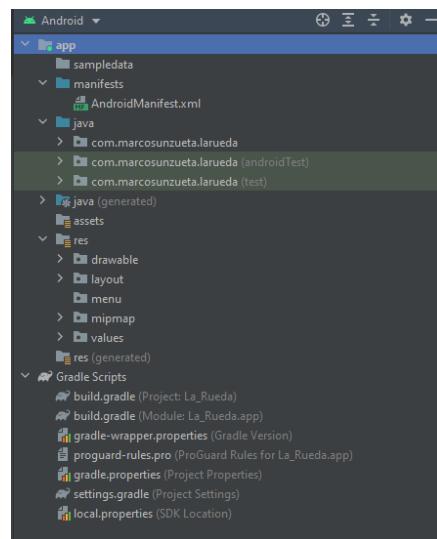


Ilustración 81: Estructura general de la aplicación

### 5.6.1. Manifests

La carpeta de manifests únicamente contiene el fichero `AndroidManifest.xml` que posee la información esencial de la aplicación para las herramientas de creación de Android, el sistema operativo Android y Google Play.

Utiliza XML para definir, entre otras cosas, las actividades de la aplicación y su estilo, algunos datos de configuración, los permisos que necesita del usuario y las direcciones.

En esta aplicación, en concreto, ha sido necesario darle permisos para que pueda escribir y leer en el almacenamiento externo del dispositivo móvil. (Android, <https://developer.android.com>, s.f.)

### 5.6.2. Java

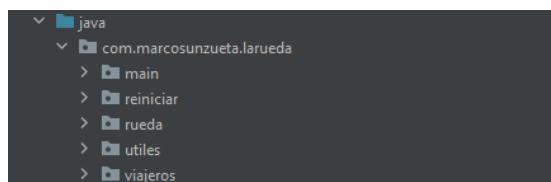


Ilustración 82: Estructura carpeta Java

- **main:**

En la carpeta Main se encuentran los ficheros que se encargan del funcionamiento del menú principal de la aplicación.

- **reiniciar:**

En la carpeta Reiniciar se encuentran los ficheros que se encargan de resetear los datos de la aplicación .

- **rueda:**

En la carpeta Rueda se encuentran los ficheros que se encargan de calcular el horario de rueda óptimo y de generar el PDF.

- **utiles:**

Carpeta en la que se encuentran una serie de ficheros que pueden ser útiles para el desarrollo de la aplicación.

- **viajeros:**

En la carpeta Rueda se encuentran los ficheros que se encargan de la gestión de los datos de los viajeros.

### 5.6.3. Res

Esta carpeta se compone de archivos arbitrarios que se almacenan sin procesar. (android, s.f.)

Este directorio se dividirá en otras subcarpetas, que serán específicas de imágenes, menús, archivos XML, estilos y otros elementos que son utilizados en la aplicación por el resto de los ficheros.

- **Drawable:**

Contiene elementos gráficos que se utilizan en la aplicación. (edu, s.f.)

En este caso se almacenan las imágenes, algunas construcciones xml utilizadas en el menú principal y bordes que se utilizan en algunas vistas de la aplicación como EditTexts, Modales o RecyclerViews.

- **Layout:**

Contiene los ficheros XML que harán las veces de vista en la arquitectura Modelo-Vista-Controlador y definen la interfaz del usuario (android, s.f.).

- **Mipmap:**

Contiene los iconos de la aplicación en distintas resoluciones y tamaños. (android, s.f.)

- **Values:**

Archivos de tipo XML que contiene valores simples como cadenas de caracteres, enteros o colores.

### 5.6.4. Gradle

Contiene los datos que utilizará el proyecto para compilar, entre otras cosas, las dependencias que importan las bibliotecas utilizadas, la mínima versión que soportará la aplicación o la versión de SDK de Android usada para compilar.

## 5.7. Instalación

Es importante tener en cuenta, que el Trabajo de Fin de Máster desarrollado, se trata de una aplicación para el sistema operativo Android, y, por tanto, la instalación será análoga a la de cualquier app descargable en Google Play Store. En primer lugar, se abre el mercado de aplicaciones de Android y se busca la app por su nombre “La rueda”, una vez localizada se descarga y el dispositivo móvil la instala en el sistema.

Esta aplicación, al tratarse de un proyecto universitario y no tener fines lucrativos, no ha sido publicada en Google Play Store, por tanto, el método de descarga e instalación ha de ser otro:

- Se puede conectar el dispositivo móvil en el que se pretende instalar la aplicación al ordenador que cuenta con el desarrollo del software en Android Studio y lanzar la app directamente al smartphone.

Se puede generar desde el propio entorno de desarrollo un APK y compartirlo mediante cualquier método al dispositivo móvil que desea tener la aplicación instalada.

## 5.8. Funcionalidad de la aplicación

Se muestran las pantallas principales de la aplicación y se explica su uso (la funcionalidad de la aplicación se recoge ampliamente en el Anexo 6).

### 5.8.1. Menú principal

Según se inicia la aplicación, esta será la vista que tenga el usuario.

El menú principal permite acceder a todas las funcionalidades de la aplicación desarrollada.



Ilustración 83: Menú principal de la aplicación

### 5.8.2. Viajeros

La apariencia del apartado de viajeros varía en función de los datos que hay almacenados en el sistema.

En este apartado se muestran los viajeros guardados y permite la gestión de estos.

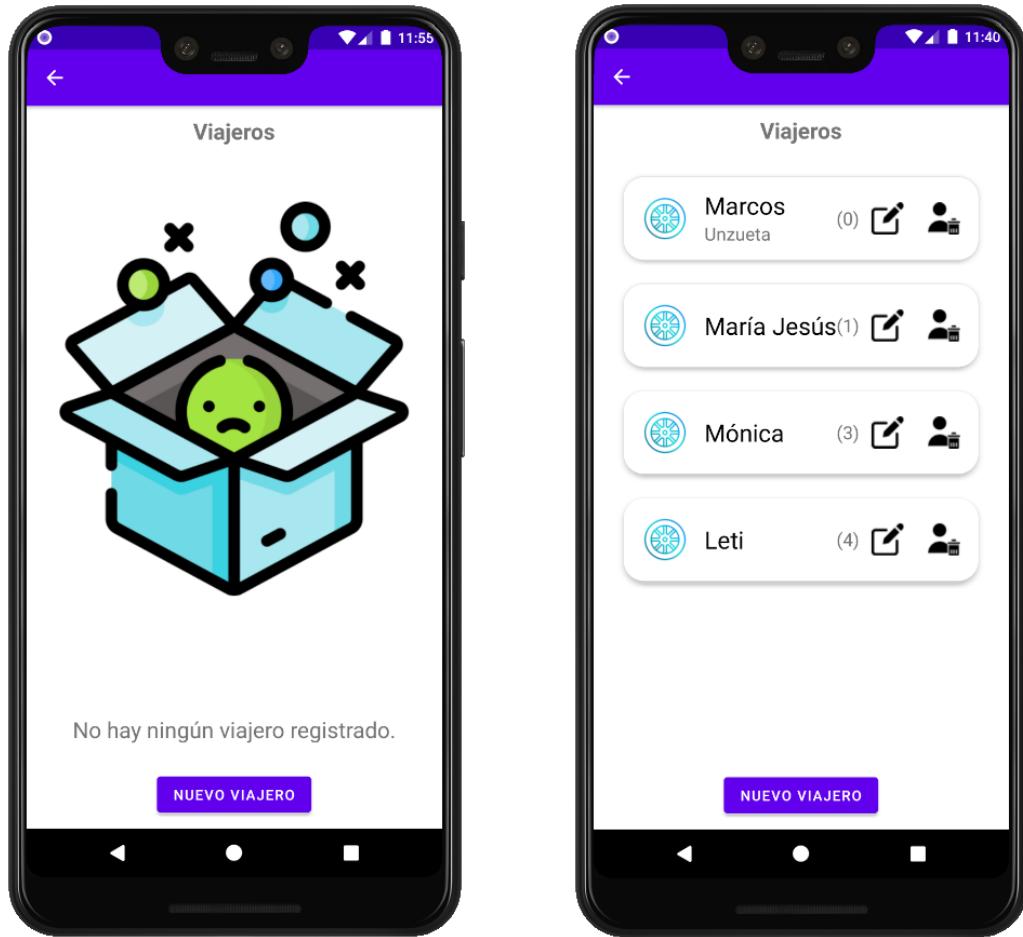


Ilustración 84: Gestión de viajeros 1

Ilustración 85: Gestión de viajeros 2

### 5.8.3. Eliminar viajero

Una vez hay viajeros almacenados, es posible eliminar a cualquiera pulsando el botón con forma de papelera que aparece a la derecha de su nombre.

Una vez se ha pulsado este botón, se muestra un modal al usuario en el que se le indica el viajero a eliminar y se confirma si desea borrarlo o no.

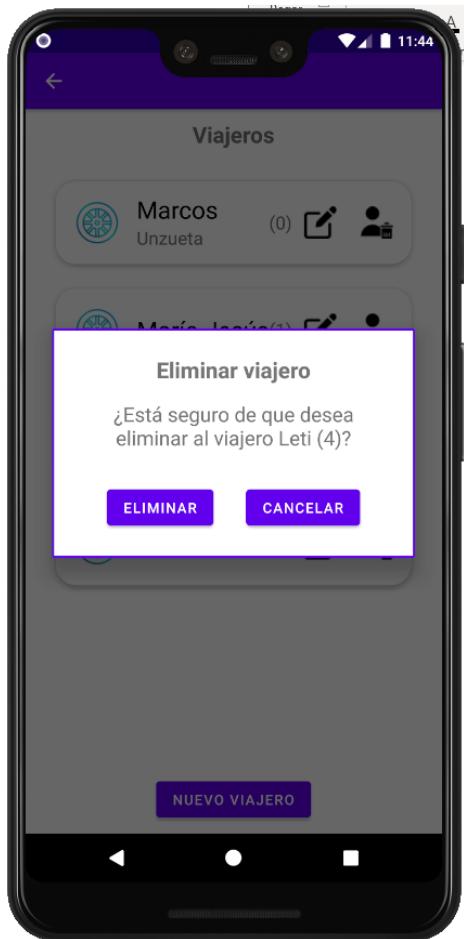


Ilustración 86: Eliminar viajero

#### 5.8.4. Añadir viajero

Desde la sección de viajeros, pulsando el botón nuevo viajero, se puede añadir uno nuevo en el almacenamiento del sistema.

Una vez pulsado el botón, se muestra el formulario que solicitará los datos necesarios para almacenar al viajero que el usuario desea.

**Nuevo Viajero**

**\*Nombre:** Nuevo

**Primer apellido:** Viajero

**Segundo apellido:** Prueba

**\*Lunes:**  
Entra: (Hora) 01 : (Mins) 00  
Sale: (Hora) 03 : (Mins) 00  
 No trabaja este día.

**\*Martes:**  
Entra: (Hora) 02 : (Mins) 00  
Sale: (Hora) 04 : (Mins) 00  
 No trabaja este día.

**\*Miércoles:**

**\*Viernes:**  
Sale: (Hora) XX : (Mins) XX  
 No trabaja este día.

**\*Sábado:**  
Entra: (Hora) XX : (Mins) XX  
Sale: (Hora) XX : (Mins) XX  
 No trabaja este día.

**\*Domingo:**  
Entra: (Hora) XX : (Mins) XX  
Sale: (Hora) XX : (Mins) XX  
 No trabaja este día.

**AÑADIR**

Ilustración 87: Nuevo viajero 1

Ilustración 88: Nuevo viajero 2

Cuando se pulsa el botón de añadir se guardarán los datos del viajero en el almacenamiento interno del sistema.

### 5.8.5. Modificar viajero

En la lista de viajeros, si se pulsa en el botón de editar de cualquier elemento, se accede al mismo formulario que en el caso de añadir viajero, pero con los datos del viajero a modificar ya puestos.

Se pulsa el botón modificar y los datos correspondientes al viajero cambian en función de los datos introducidos en el formulario por el usuario.

### 5.8.6. Generar rueda

La apariencia del apartado de rueda varía en función de los datos que hay almacenados en el sistema.

Esta sección permite realizar todas las acciones relacionadas con el “horario de rueda”.

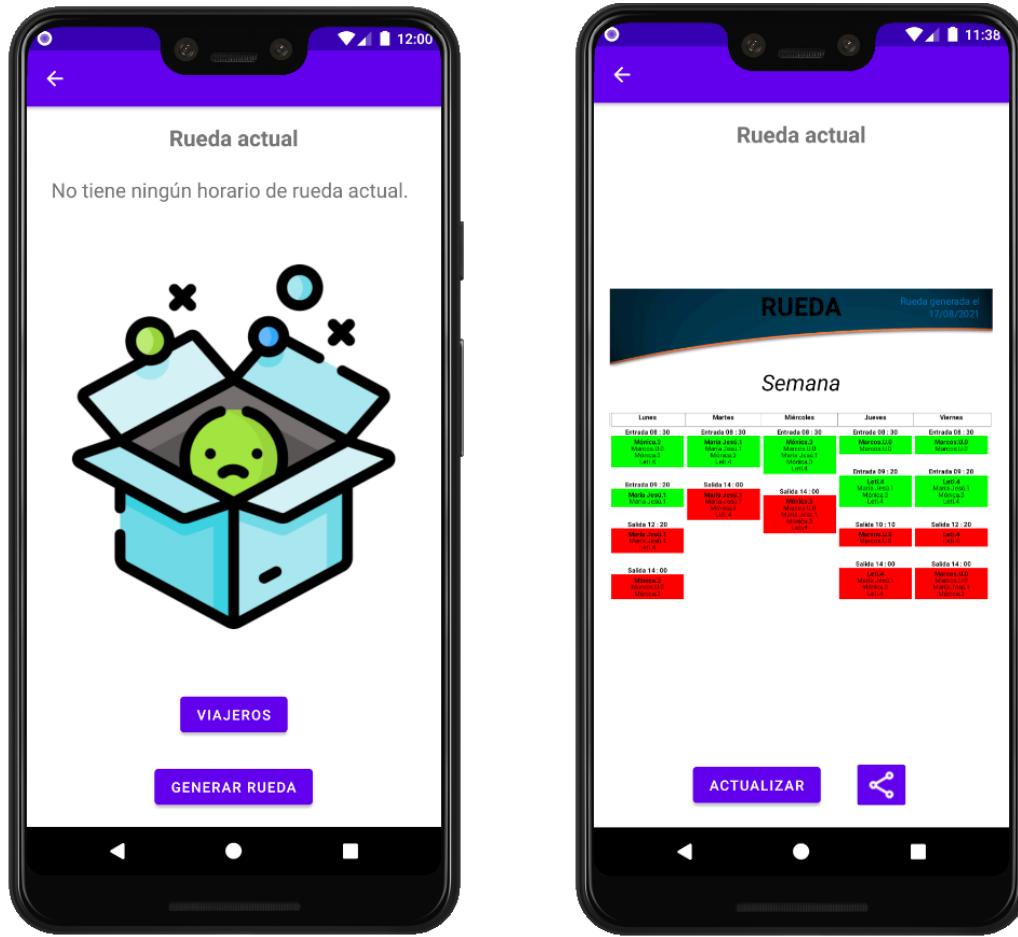


Ilustración 89: Gestión de rueda sin horario

Ilustración 90: Gestión de rueda con horario generado

El PDF generado se puede ampliar en la aplicación para mejorar su visibilidad.



Ilustración 91: Horario ampliado

#### 5.8.7. Compartir rueda

Cuando se pulsa el botón de compartir, la aplicación muestra todos los medios que tiene para exportar el PDF y brinda al usuario la opción de seleccionar el que desee.



Ilustración 92: Compartir horario

#### 5.8.8. Reiniciar rueda

Si desde el menú principal se pulsa el botón de reiniciar, se muestra al usuario el modal correspondiente a reiniciar los datos de la aplicación rueda.



Ilustración 93: Reiniciar rueda

Si se pulsa al botón eliminar del modal se borran del sistema los datos de viajeros y los datos de rueda.

## 6. Conclusiones y líneas de trabajo futuro

En este apartado se recogerán las conclusiones que han surgido del propio desarrollo del proyecto una vez este ha finalizado.

Para terminar, se comentarán unas posibles mejoras para desarrollarlas en un futuro.

### 6.1. Conclusiones

Con el proyecto ya realizado, llega la hora de rebobinar y analizar los objetivos que se definieron al principio del desarrollo, y que se encuentran recogidos en esta memoria.

El objetivo principal, que consistía en desarrollar un software para optimizar los viajes por trabajo generando calendarios óptimos, para que un grupo de personas que desean desplazarse de manera conjunta pueda hacerlo eficientemente, respetando los horarios laborales de los viajeros, minimizando el tiempo que estos están innecesariamente en el centro de trabajo y minimizando el número de coches desplazados semanalmente, ha sido llevado a cabo con éxito, habiendo creado una aplicación para dispositivos móviles que aúna todas estas restricciones y características.

Se procede, también, a enumerar los objetivos de grano más fino, justificando en todo caso por qué se considera que dicho objetivo se ha cumplido.

#### 6.1.1. Objetivos funcionales

##### - Gestión de viajeros:

Se ha realizado con éxito, consiguiendo que los usuarios puedan gestionar correctamente todos los viajeros que participarán en el horario de rueda.

Se muestran los viajeros almacenados en una lista, se permite modificar los datos de cada uno de ellos, se permite eliminarlos de forma permanente del almacenamiento interno del sistema y añadir nuevos.

Todas estas operaciones realizadas sobre los viajeros también implican su efecto análogo sobre el horario de trabajo de estos.

##### - Calcular todos los “horarios de rueda” posibles

Se ha realizado un algoritmo que, a partir de todos los datos almacenados de los viajeros y, por tanto, de sus horarios obtenga todos los “horarios de rueda” posibles.

Este algoritmo se ejecuta en segundo plano para no sobrecargar el plano principal de la aplicación y divide su trabajo en hilos, tratando de optimizar el procesamiento.

Los horarios contienen los coches que se desplazan cada día de la semana, los conductores de estos, los viajeros que se desplazan a la ida, los viajeros que se desplazan a la vuelta, la hora de entrada y la hora de salida entre otros datos.

##### - Evaluar todos los horarios de rueda generados

Se ha realizado un método que permite calcular un valor, denominado bondad, correspondiente a cada uno de los horarios generados en el algoritmo mencionado anteriormente y que representa lo bueno que es ese horario.

Este método pondera el número de coches desplazados al centro de trabajo y las horas esperadas por el viajero en el centro, dando como resultado un valor numérico.

- **Seleccionar el mejor resultado**

Se ha llevado a cabo un proceso de selección sobre todos los horarios calculados teniendo en cuenta la evaluación mencionada en el apartado anterior.

- **Generar un fichero exportable con el mejor resultado**

El sistema es capaz de generar un fichero PDF, perfectamente interpretable por un usuario humano, que contenga todos los datos del horario de rueda óptimo (calculado, evaluado y seleccionado con los métodos ya mencionados):

La aplicación permite al usuario compartir el fichero haciendo uso de aplicaciones de terceros como WhatsApp, Telegram o Gmail.

- **Reiniciar el sistema**

Se ha desarrollado una opción de reinicio rápido que permite al usuario devolver al sistema al estado inicial con tan solo pulsar un botón.

#### **6.1.2. Objetivos no funcionales**

- **Accesibilidad:**

Se ha conseguido favorecer la accesibilidad realizando una aplicación para dispositivos móviles Android (el dispositivo que más usuarios tiene y su sistema operativo más común) cuyo uso resulte sencillo e intuitivo.

- **Funcionamiento en Android**

La aplicación funciona perfectamente en los dispositivos móviles con sistema operativo Android.

- **Adaptabilidad y escalabilidad**

Se ha conseguido utilizando programación orientada a objetos, reutilizando gran cantidad de métodos y clases y comentando con mucho detalle el código de la aplicación.

- **Interfaz amigable y sencilla**

La interfaz realizada es muy intuitiva y sencilla, permitiendo realizar acciones complejas en muy pocas transiciones, explicando el método de uso y ofreciendo una funcionalidad ceñida al usuario objetivo.

- **Eficacia**

Como se ha mencionado desde un principio, esta aplicación obtiene siempre como resultado el mejor horario dentro de todas las opciones posibles.

## **6.2. Aprendizaje**

Siempre un alumno se enfrenta a un proyecto de estas dimensiones y características, conviene hacer un análisis del aprendizaje obtenido a lo largo del desarrollo de este.

Este aprendizaje se ve marcado por la gran cantidad de tecnologías y herramientas informáticas usadas, así como los principios matemáticos que se utilizan.

#### **6.2.1. Android Studio :**

El alumno ha continuado su aprendizaje referente a la realización de aplicaciones móviles con el entorno de desarrollo Android Studio.

Este proyecto ha utilizado herramientas del IDE que el alumno desconocía, enriqueciendo así su manejo del software y convirtiéndole en un mejor programador.

#### **6.2.2. Automatización PDF:**

El alumno ha aprendido a automatizar el desarrollo de PDFs utilizando bibliotecas de Java.

La curva de aprendizaje ha sido relativamente lenta ya que el proceso de automatización de estos ficheros se realiza píxel a píxel siendo bastante tedioso y costoso, no obstante, gracias a los conocimientos adquiridos, en desarrollos posteriores se podrá realizar en menos tiempo.

#### **6.2.3. Cálculos matemáticos:**

Está claro que la parte más compleja de esta aplicación es el cálculo del horario óptimo.

Los algoritmos realizados para calcular todos los horarios en la mínima cantidad de operaciones posibles han costado mucho trabajo.

El alumno ha aprendido a idear algoritmos difíciles y depurarlos.

#### **6.2.4. Uso SQLite:**

El alumno nunca había usado una base de datos empotrada en la aplicación para un proyecto de estas dimensiones, por tanto, ha aprendido muchos conceptos de gestión, creación y manejo de esta.

#### **6.2.5. Paralelización del trabajo y segundo plano:**

El alumno ha utilizado los conceptos obtenidos en varias asignaturas del máster para paralelizar correctamente el trabajo a realizar y conseguir así que la gran cantidad de operaciones realizadas lleven menos tiempo.

Por otra parte se ha decidido realizar este procesamiento más pesado en un segundo plano, librando al plano principal de la aplicación del peso computacional.

#### **6.2.6. Documentación:**

El proceso de documentación del proyecto desarrollado ha sido muy enriquecedor y ha ayudado al alumno a ordenar varios de los conocimientos adquiridos a lo largo de su educación en la informática.

### **6.3. Trabajo futuro**

El proyecto realizado tiene una gran cantidad de aristas y posibilidades a explotar.

Si bien, el cálculo y la generación de los horarios era un reto bastante ambicioso para acometer en el tiempo estimado para el desarrollo de un Trabajo de Fin de Máster, el resultado obtenido y desarrollado puede ampliarse en un futuro.

Una opción sería realizar una especie de red social con la aplicación, en la que se crea un espacio de trabajo común y cada invitado es el encargado de añadir sus datos y modificaciones que, automáticamente, actualizarán el horario.

Si el alumno dispusiera de un servidor se podría realizar todo el proceso de cálculo en él, haciendo que se envíe el resultado al correo del usuario y evitando la espera con la aplicación encendida en el teléfono.

Sería un gran avance permitir el cálculo del coste del viaje en función del origen y el destino, así como añadir la posibilidad de que ciertas personas sean recogidas de camino. Esta mejora requerirá mucho tiempo de desarrollo.

Está pensado obtener una versión de la aplicación para dispositivos móviles con sistema operativo IOS.

# Referencias

- (BABOK), B. A. (s.f.). <https://www.iiba.org/>. Obtenido de <https://www.iiba.org/career-resources/business-analysis-professionals-foundation-for-success/babok/>
- Andalucía, J. d. (s.f.). <http://www.juntadeandalucia.es>. Obtenido de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/407>
- Android. (s.f.). <https://developer.android.com>. Obtenido de <https://developer.android.com/jetpack/androidx>
- Android. (s.f.). <https://developer.android.com>. Obtenido de <https://developer.android.com/studio/intro?hl=es-419#:~:text=Android%20Studio%20te%20ayuda%20a%20depurar%20y%20mejorar%20el%20rendimiento,depuraci%C3%B3n%20y%20an%C3%A1lisis%20de%20rendimiento>.
- Android. (s.f.). <https://developer.android.com>. Obtenido de <https://developer.android.com/studio/intro?hl=es-419#:~:text=Android%20Studio%20te%20ayuda%20a%20depurar%20y%20mejorar%20el%20rendimiento,depuraci%C3%B3n%20y%20an%C3%A1lisis%20de%20rendimiento>.
- Android. (s.f.). <https://developer.android.com>. Obtenido de <https://developer.android.com/guide/topics/manifest/manifest-intro?hl=es-419#:~:text=Todos%20los%20proyectos%20de%20apps,operativo%20Android%20y%20Google%20Play>
- android, D. (s.f.). <https://developer.android.com>. Obtenido de <https://developer.android.com/guide/topics/resources/providing-resources?hl=es-419>
- androidauthority. (s.f.). <https://www.androidauthority.com>. Obtenido de <https://www.androidauthority.com/use-sqlite-store-data-app-599743/>
- androidcurso. (s.f.). <http://www.androidcurso.com>. Obtenido de <http://www.androidcurso.com/index.php/465>
- barteksc. (s.f.). <https://github.com/barteksc>. Obtenido de <https://github.com/barteksc/AndroidPdfViewer>
- Blanco, M. (s.f.). <https://wikis.fdi.ucm.es>. Obtenido de [https://wikis.fdi.ucm.es/ELP/Especificaci%C3%B3n\\_de\\_Requisitos\\_Software\\_seg%C3%BAn\\_el\\_est%C3%A1ndar\\_IEEE\\_830](https://wikis.fdi.ucm.es/ELP/Especificaci%C3%B3n_de_Requisitos_Software_seg%C3%BAn_el_est%C3%A1ndar_IEEE_830)
- Calabar, P. (s.f.). [www.dc.fi.udc.es](http://www.dc.fi.udc.es). Obtenido de <https://www.dc.fi.udc.es/~cabalar/md/t6.pdf>
- campusmpv. (s.f.). [www.campusmpv.es](http://www.campusmpv.es). Obtenido de <https://www.campusmpv.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>
- carpling. (s.f.). [www.carpling.com](http://www.carpling.com). Obtenido de <http://www.carpling.com/es/car/compartir-coche-al-trabajo/>
- Caules, C. Á. (2015). <https://www.arquitecturajava.com>. Obtenido de <https://www.arquitecturajava.com/que-es-gradle/>
- Confidencial, E. (s.f.). [www.elconfidencial.com](http://www.elconfidencial.com). Obtenido de [https://www.elconfidencial.com/economia/2020-02-16/gran-migracion-diaria-trabajadores-provincias-epa\\_2453516/](https://www.elconfidencial.com/economia/2020-02-16/gran-migracion-diaria-trabajadores-provincias-epa_2453516/)

- Cuajimalpa, U. (s.f.). <http://www.cua.uam.mx/>. Obtenido de [http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas\\_Analisis\\_Requerimiento.pdf](http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_Analisis_Requerimiento.pdf)
- dedocar. (s.f.). <https://dedocar.org/>. Obtenido de <https://dedocar.org/>
- edu. (s.f.). <https://stuff.mit.edu/>. Obtenido de <https://stuff.mit.edu/afs/sipb/project/android/docs/guide/topics/resources/drawable-resource.html>
- esan, C. (2018). <https://www.esan.edu.pe>. Obtenido de <https://www.esan.edu.pe/apuntes-empresariales/2018/10/microsoft-project-su-aplicacion-en-la-gestion-de-proyectos/>
- Esteban. (2020). <https://www.elespanol.com>. Obtenido de [https://www.elespanol.com/elandroidelibre/tutoriales/20200117/android-studio-puedes-hacer-programa/460455295\\_0.html](https://www.elespanol.com/elandroidelibre/tutoriales/20200117/android-studio-puedes-hacer-programa/460455295_0.html)
- exactas, a. (s.f.). <http://www.aulavirtual-exactas.dyndns.org>. Obtenido de <http://www.aulavirtual-exactas.dyndns.org/claroline/backends/download.php?url=L2NhDE0aF9Nb2RDb25jZXBEB21pbmlvLnBkZg%3D%3D&cidReset=true&cidReq=2014ANIN2>
- Grial. (s.f.). <https://repositorio.grial.eu>. Obtenido de <https://repositorio.grial.eu/bitstream/grial/1948/1/7.%20PU-2020.pdf>
- ima. (s.f.). <http://ima.udg.edu>. Obtenido de <http://ima.udg.edu/~sellares/einf-es2/present1011/metodopesadesrup.pdf>
- journify. (s.f.). <https://journify.com/>. Obtenido de <https://journify.com/>
- juntadeandalucia. (s.f.). <https://www.juntadeandalucia.es>. Obtenido de <https://www.juntadeandalucia.es/servicios/madeja/contenido/procedimiento/20#:~:text=Requisitos%20de%20informaci%C3%B3n%2C%20describen%20la,cumplimiento%20debe%20velar%20el%20sistema>.
- manuel.cillero. (s.f.). <https://manuel.cillero.es>. Obtenido de <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-despliegue/>
- material.io. (s.f.). <https://material.io>. Obtenido de <https://material.io/components/sliders/android>
- miReal. (s.f.). <https://mireal.me>. Obtenido de <https://mireal.me/que-es-android-x-porque-como-convertir-un-proyecto-de-android-studio-existente-a-android-x-u4universo/>
- Montenegro, A. (2021). <https://www.elmundo.es>. Obtenido de <https://www.elmundo.es/ciencia-y-salud/medio-ambiente/2021/06/21/60cca5c3fc6c83672b8b45b3.html>
- Muradas, Y. (s.f.). <https://openwebinars.net>. Obtenido de <https://openwebinars.net/blog/que-es-gradle/>
- Page, M. (s.f.). [www.michaelpage.es](http://www.michaelpage.es). Obtenido de <https://www.michaelpage.es/prensa-estudios/estudios/transport-commute>
- Plaza, V. (2019). <https://valenciaplaza.com>. Obtenido de <https://valenciaplaza.com/journify-compartir-coche-universidades-parques-tecnologicos>
- qastack. (s.f.). <https://qastack.mx>. Obtenido de <https://qastack.mx/programming/51280090/what-is-androidx>
- Rómmel, F. (s.f.). <https://blog.escuelactec.com>. Obtenido de <https://blog.escuelactec.com/que-es-sqlite/>

- rploaiza. (s.f.). <https://es.slideshare.net/rploaiza>. Obtenido de <https://es.slideshare.net/rploaiza/vistas-arquitectonicas-ingenieria-de-software>
- scalegrid. (s.f.). <https://scalegrid.io>. Obtenido de <https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/>
- Sevilla, U. d. (s.f.). <http://www.lsi.us.es>. Obtenido de [http://www.lsi.us.es/~javierj/cursos\\_ficheros/metricaUML/CasosUsoUML.pdf](http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/CasosUsoUML.pdf)
- statista. (s.f.). <https://es.statista.com>. Obtenido de <https://es.statista.com/grafico/16580/lenguajes-de-programacion-mas-usados-del-mundo/>
- vpnmentor. (s.f.). <https://es.vpnmentor.com>. Obtenido de <https://es.vpnmentor.com/blog/tendencias-de-internet-estadisticas-y-datos-en-los-estados-unidos-y-el-mundo/>
- xataka. (s.f.). <https://www.xataka.com>. Obtenido de <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>
- Xie, Y. M. (2020). <https://yiminshum.com>. Obtenido de <https://yiminshum.com/mobile-movil-app-2020/>
- Ybrceyo. (s.f.). <http://1.bp.blogspot.com>. Obtenido de [http://1.bp.blogspot.com/-UhmlYbrceyo/Vgx6xLbk6kI/AAAAAAA06I/9hk\\_PMfvF5w/s1600/Android%2BVirtual%2BDevice%2BManager%2BNexus%2B6P.png](http://1.bp.blogspot.com/-UhmlYbrceyo/Vgx6xLbk6kI/AAAAAAA06I/9hk_PMfvF5w/s1600/Android%2BVirtual%2BDevice%2BManager%2BNexus%2B6P.png)