



## Trabajo de Fin de Grado

Grado en Ingeniería Informática

Ingeniería de Software

---

### Prototipo de Sistema de Peaje - auToll

---

*Konstantin Todorov Andreev*

#### Dirección

Maider Azanza  
Ekaitz Jauregi

28 de octubre de 2023



# **Agradecimientos**

Quiero expresar mi sincero agradecimiento a mis tutores y directores de proyecto, Maider Azanza y Ekaitz Jauregi, por su inestimable guía, constante apoyo y motivación a lo largo de todo el proceso de desarrollo de este proyecto. Sin su orientación y compromiso, este proyecto no habría sido posible.

Además, deseo expresar mi profunda gratitud a mi familia y amigos por su apoyo inquebrantable y comprensión.



# Resumen

Este documento presenta un escenario hipotético en el contexto de un Trabajo Fin de Grado. En esta simulación, se plantea la necesidad de reemplazar los sistemas de peaje existentes por uno más eficiente, una iniciativa que despierta el interés de la DGT y el Gobierno Español. La propuesta se centra en el diseño y desarrollo de un nuevo sistema de gestión y cobro vial que emplea tecnologías avanzadas como el reconocimiento de matrículas.

Es importante enfatizar que este proyecto se enfoca en el análisis, investigación y desarrollo de sistemas de tecnologías de alto nivel en el campo de la ingeniería de software, explorando soluciones avanzadas para desafíos en este ámbito.

En colaboración con la Universidad del País Vasco UPV/EHU, se planea llevar a cabo este proyecto. El sistema que se desarrollará simulará el funcionamiento de un sistema real, donde "cámaras" registrarán automáticamente las matrículas de los vehículos, gestionando los cobros correspondientes por el uso de la autopista.

Además, se contempla la implementación de una interfaz de usuario que permitirá a los usuarios acceder a su historial de uso, revisar información de facturación, realizar pagos y resolver cualquier problema que pueda surgir durante el proceso.

Es fundamental destacar que este escenario es puramente hipotético y carece de vínculos con implementaciones o situaciones reales. El sistema que se desarrollará será una simulación simplificada en comparación con un sistema de peaje real debido a las limitaciones de recursos disponibles.



# Índice de contenidos

<b>Índice de contenidos</b>	<b>v</b>
<b>Índice de figuras</b>	<b>x</b>
<b>Índice de tablas</b>	<b>xii</b>
<b>1 Visión General</b>	<b>1</b>
1.1. Contexto y Justificación . . . . .	1
1.2. Público Objetivo . . . . .	2
1.3. Visión del Proyecto . . . . .	2
<b>2 Planificación</b>	<b>5</b>
2.1. Alcance del Proyecto . . . . .	5
2.1.1. Descripción del Proyecto Prototipo Virtual . . . . .	5
2.1.2. Objetivos concretos del Proyecto . . . . .	5
2.1.3. Objetivos secundarios del Proyecto . . . . .	6
2.1.4. Requisitos del Proyecto . . . . .	6
2.1.5. Marco Legal . . . . .	6
2.1.6. Licencia del Proyecto . . . . .	7
2.2. Descripción de las exclusiones y supuestos . . . . .	7
2.2.1. Exclusiones . . . . .	7
2.2.2. Supuestos . . . . .	8
2.2.3. Entradas al proyecto . . . . .	8
2.2.4. Salidas del proyecto . . . . .	8
2.3. Ciclo de Vida del Proyecto . . . . .	8
2.4. Estructura de Desglose de Trabajo (EDT) . . . . .	11
2.4.1. Periodos de realización de las tareas y fechas de terminación de los entregables . . . . .	14
2.4.2. Dependencias entre las tareas . . . . .	15
2.5. Estimación de dedicación a cada una de las tareas . . . . .	17
2.6. Caracterización del sistema de información y comunicaciones en el proyecto	19
2.6.1. Estructura del proyecto . . . . .	19
2.6.2. Denominaciones . . . . .	19
2.6.3. Copias de Seguridad . . . . .	20
2.6.4. Comunicaciones . . . . .	21
2.7. Análisis de los riesgos . . . . .	21

<b>3 Captura de Requisitos</b>	<b>25</b>
3.1. Stakeholders y Usuarios . . . . .	25
3.1.1. Identificación de Stakeholders . . . . .	25
3.1.2. Identificación de los actores del sistema . . . . .	25
3.1.3. Requisitos Funcionales . . . . .	26
3.1.4. Requisitos No Funcionales . . . . .	27
3.2. Casos de Uso . . . . .	28
3.2.1. Diagrama de Casos de Uso del modulo IPD . . . . .	28
3.3. Diagrama de Dominio . . . . .	31
3.3.1. Entidades del Dominio . . . . .	31
3.3.2. Relaciones . . . . .	33
3.4. Diseño de la Base de Datos . . . . .	34
3.4.1. Modelo de Datos . . . . .	34
3.4.2. Normalización . . . . .	34
<b>4 Herramientas y Tecnologías</b>	<b>37</b>
4.1. Herramientas . . . . .	37
4.1.1. IntelliJ IDEA . . . . .	37
4.1.2. IntelliJ DataGrip . . . . .	37
4.1.3. IntelliJ WebStorm . . . . .	37
4.1.4. VSCode . . . . .	38
4.1.5. MySQL Workbench . . . . .	38
4.2. Aplicación Web . . . . .	38
4.2.1. Experiencia del usuario en las aplicaciones web . . . . .	38
4.2.2. Consideraciones de rendimiento en términos de velocidad y recursos . . . . .	38
4.2.3. Aspectos de seguridad . . . . .	38
4.2.4. Selección del enfoque más adecuado para satisfacer los objetivos del proyecto. . . . .	39
4.3. REST vs SOAP . . . . .	39
4.3.1. Comparación de REST y SOAP . . . . .	39
4.3.2. Principales diferencias entre REST y SOAP . . . . .	40
4.3.3. Consideraciones de rendimiento y seguridad . . . . .	40
4.3.4. Selección del protocolo adecuado para el proyecto . . . . .	41
4.4. Front-end . . . . .	41
4.4.1. Uso de JavaScript, HTML, CSS o Bootstrap y Thymeleaf . . . . .	41
4.4.2. React vs Angular . . . . .	41
4.4.3. Ejemplos de casos de uso para React y Angular . . . . .	42
4.4.4. Selección del Framework Adecuado para el Proyecto . . . . .	43
4.5. Back-End . . . . .	43
4.5.1. Uso de Java en el proyecto . . . . .	43
4.5.2. Uso de Spring Framework en el proyecto . . . . .	44
4.5.3. Distintos proyectos Spring . . . . .	44
4.5.4. Configuración y módulos relevantes de Spring . . . . .	44
4.6. MySQL Base de Datos . . . . .	44
4.7. Python . . . . .	44
4.7.1. Áreas específicas donde se utilizará Python . . . . .	44
4.7.2. Razones para considerar el uso de Python en el proyecto . . . . .	45

4.8.	Snapshot API de Plate Recognizer . . . . .	45
4.8.1.	Descripción de la Snapshot API de Plate Recognizer . . . . .	45
4.8.2.	Uso de la API para reconocimiento de matrículas . . . . .	45
4.8.3.	Beneficios y aplicaciones de Plate Recognizer en el proyecto . . . . .	45
4.8.4.	Desventajas de Plate Recognizer en el proyecto . . . . .	45
4.9.	Testeo de Código: Postman, JUnit y Mockito . . . . .	46
4.9.1.	JUnit y AssertJ . . . . .	46
4.9.2.	Mockito . . . . .	46
4.9.3.	Uso de Postman en el proyecto . . . . .	46
4.10.	Logging: Log4j2 . . . . .	47
4.11.	Project Lombok . . . . .	47
4.12.	DevOps . . . . .	47
4.12.1.	Jenkins . . . . .	47
4.12.2.	Docker . . . . .	48
<b>5</b>	<b>Arquitectura del Sistema</b>	<b>49</b>
5.1.	Conjunto de tecnologías utilizadas . . . . .	49
5.2.	Modelo de flujo de datos del sistema . . . . .	50
5.3.	Capas del sistema . . . . .	51
5.4.	CI/CD pipeline con Maven y Docker . . . . .	53
5.5.	Configuración de Producción y Desarrollo . . . . .	54
5.6.	Seguridad en el sistema . . . . .	54
5.6.1.	Seguridad de Usuario y Simulador . . . . .	54
5.6.2.	Seguridad de los servicios REST . . . . .	55
5.6.3.	K-Anonimización . . . . .	55
5.7.	Cambios en la estructura del proyecto . . . . .	55
5.8.	Lecciones aprendidas . . . . .	58
5.8.1.	Creación de CI/CD Pipeline con Jenkins en Docker . . . . .	58
<b>6</b>	<b>Interfaz de Procesamiento de Datos - IPD (Eagle)</b>	<b>61</b>
6.1.	Preparación de Datos . . . . .	61
6.1.1.	Puntos de control . . . . .	62
6.1.2.	Tasas . . . . .	62
6.1.3.	Secciones . . . . .	63
6.2.	Incremento de Base de Datos I3 . . . . .	64
6.3.	Relación con los casos de uso IPD . . . . .	65
6.4.	Capa de Servicio de la IPD . . . . .	66
6.4.1.	Simulador IPD . . . . .	66
6.4.2.	Servicios RESTful . . . . .	67
6.4.3.	Reconocimiento de Matrícula . . . . .	67
6.5.	Seguridad de la IPD . . . . .	67
<b>7</b>	<b>Software de Visualización de Datos - SVD</b>	<b>69</b>
7.1.	Incremento de Base de Datos I3 . . . . .	69
7.2.	Relación con los Casos de Uso . . . . .	70
7.3.	Diseño de la aplicación Web . . . . .	71
7.4.	Seguridad . . . . .	72
7.4.1.	Autenticación de los usuarios . . . . .	73

7.5. Elemento de búsqueda . . . . .	74
7.6. Captura de errores . . . . .	75
7.7. Panel de control . . . . .	76
7.8. Tramos . . . . .	77
7.9. Generación JWT para nuestra API . . . . .	77
<b>8 Visión por Computadora</b>	<b>79</b>
8.1. Algoritmo de Reconocimiento . . . . .	79
8.1.1. OpenCV y Tesseract . . . . .	79
8.1.2. Etapas del Algoritmo de Reconocimiento de Matrículas . . . . .	80
8.1.3. Demostración del Algoritmo . . . . .	80
8.2. Captura de las Etiquetas Ambientales . . . . .	80
8.2.1. Web Scraping con Scrapy . . . . .	80
8.2.2. Resultados del Web Scraping . . . . .	82
<b>9 Pruebas del Sistema</b>	<b>83</b>
9.1. Pruebas de Servicios . . . . .	83
9.1.1. Pruebas Unitarias . . . . .	83
9.1.2. Pruebas de Integración . . . . .	85
9.2. Métricas y Verificación de Requisitos . . . . .	87
<b>10 Seguimiento y Control</b>	<b>89</b>
10.1. Control de Cambios . . . . .	89
10.2. Retrasos significativos . . . . .	91
10.2.1. Identificación de Retrasos . . . . .	91
10.2.2. Causas de los retrasos . . . . .	91
10.2.3. Impacto de los retrasos . . . . .	91
10.2.4. Acciones correctivas . . . . .	92
10.3. Replanificación del proyecto . . . . .	93
10.3.1. Replanificación 1 . . . . .	93
10.3.2. Replanificación 2 . . . . .	95
10.4. Seguimiento de las horas trabajo . . . . .	96
10.4.1. Análisis de las horas de trabajo . . . . .	96
10.5. Riesgos y mitigación . . . . .	100
10.5.1. R-5: Riesgo de Empleabilidad . . . . .	100
10.5.2. R-3: Riesgo de Tiempo . . . . .	100
<b>11 Comparación en el entorno de sistemas de alto nivel</b>	<b>101</b>
11.1. Resumen del Sistema . . . . .	101
11.1.1. CI/CD pipeline . . . . .	102
11.1.2. Servicios RESTful . . . . .	102
11.1.3. IPD simulador . . . . .	102
11.1.4. SVD Web Application . . . . .	103
11.1.5. Consideraciones de Seguridad y Privacidad . . . . .	103
<b>12 Conclusiones</b>	<b>105</b>
12.1. Logros . . . . .	105
12.2. Cumplimiento de los Objetivos . . . . .	105

12.2.1. Objetivos Principales . . . . .	105
12.2.2. Objetivos Secundarios . . . . .	105
12.3. Trabajo Futuro . . . . .	106
12.3.1. Arquitectura de Microservicios . . . . .	106
12.3.2. Base de Datos Distribuidas y Balanceo de Carga . . . . .	106
12.3.3. Mejorando el Frontend: React . . . . .	106
12.3.4. CI/CD Pipeline - Parte 2 . . . . .	106
12.3.5. Despliegue del Sistema . . . . .	106
12.4. Conclusiones . . . . .	106
<b>Apéndice</b>	<b>109</b>
<b>Actas de Reuniones</b>	<b>111</b>
Plantilla de Actas de Reuniones del Proyecto . . . . .	112
KA-TFG-AR-tt-nn-dd/mm/yyyy . . . . .	112
KA-TFG-AR-Inicion-Proyecto-00-14/07/2023 . . . . .	113
KA-TFG-AR-Seguimiento-y-Control-01-27/07/2023 . . . . .	114
KA-TFG-AR-Seguimiento-y-Control-02-05/09/2023 . . . . .	115
KA-TFG-AR-Seguimiento-y-Control-03-19/09/2023 . . . . .	116
KA-TFG-AR-Seguimiento-y-Control-04-25/09/2023 . . . . .	118
KA-TFG-AR-Seguimiento-y-Control-05-10/10/2023 . . . . .	120
KA-TFG-AR-Seguimiento-y-Control-06-16/10/2023 . . . . .	121
KA-TFG-AR-LPR-07-19/10/2023 . . . . .	122
KA-TFG-AR-Cierre-de-proyecto-08-23/10/2023 . . . . .	123
<b>Bibliografía</b>	<b>125</b>

# Índice de figuras

1.1.	Ejemplo de atasco en sistema de peaje [1] . . . . .	1
2.1.	Incrementos concretos del ciclo de vida del proyecto . . . . .	9
2.2.	Estructura de Desglose de Trabajo . . . . .	12
2.3.	Dependencias entre las tareas . . . . .	15
2.4.	Diagrama de Gantt . . . . .	17
3.1.	Diagrama de casos de uso IPD . . . . .	28
3.2.	Diagrama de casos de uso SVD . . . . .	29
3.3.	Diagrama de Dominio . . . . .	31
3.4.	Diagrama de Diseño de la Base de Datos . . . . .	34
5.1.	Conjunto de tecnologías utilizados . . . . .	49
5.2.	Modelo de flujo de datos del sistema . . . . .	50
5.3.	Capas del sistema . . . . .	52
5.4.	CI/CD pipeline . . . . .	53
5.5.	Estructura del proyecto actual . . . . .	56
5.6.	Estructura del proyecto final . . . . .	57
6.1.	Sección AP-1 . . . . .	64
6.2.	Sección AP-8 . . . . .	64
6.3.	Sección AP-68 . . . . .	64
6.4.	Cambio al Modelo de Dominio - Punto . . . . .	65
6.5.	Diagrama de Dominio incremento 3 . . . . .	66
7.1.	Diagrama de Dominio en el Incremento 4 . . . . .	70
7.2.	UI de la pagina inicial . . . . .	71
7.3.	DaoAuthenticationProvider esquema . . . . .	74
7.4.	Arbol Trie . . . . .	75
7.5.	Barra de búsqueda 1 . . . . .	75
7.6.	Barra de búsqueda 2 . . . . .	75
7.7.	Página de error . . . . .	76
7.8.	Panel de control . . . . .	76
7.9.	Tramos . . . . .	77
8.1.	Ejecución del algoritmo . . . . .	81
10.1.	Estructura de Desglose de Trabajo (EDT) expandido . . . . .	90
10.2.	Diagrama Gantt modificado, replanificación 1 . . . . .	93

## *ÍNDICE DE FIGURAS*

xI

10.3. Análisis de horas en la replanificación 2 . . . . .	95
10.4. Diagrama Gantt en la replanificación 2 . . . . .	96
10.5. Primer análisis de las horas de trabajo empleadas . . . . .	97
10.6. Segundo análisis de las horas de trabajo empleadas . . . . .	98
10.7. Último análisis de horas . . . . .	99

# Índice de tablas

2.1.	Tabla de estimación de dedicación de las tareas . . . . .	18
4.1.	Ventajas y desventajas en experiencia del usuario . . . . .	38
4.2.	Ventajas y desventajas React . . . . .	42
4.3.	Ventajas y desventajas Angular . . . . .	42
6.1.	Asociación entre Puntos y Secciones . . . . .	62
6.2.	Asociación entre Tasas y Secciones . . . . .	63
10.1.	Tabla de acciones correctivas sobre rs-1 . . . . .	92
10.2.	Tabla de estimación de dedicación de las tareas . . . . .	92
10.3.	Tabla de estimación de dedicación de las tareas . . . . .	94
10.4.	Tabla de estimación de dedicación de las tareas incremento 5 . . . . .	96

# 1

## CAPÍTULO

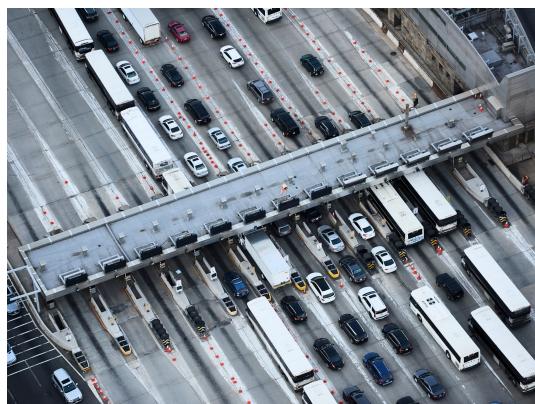
# Visión General

En este capítulo se presenta tanto la visión general del proyecto como la justificación del proyecto de cara al público objetivo y la visión del proyecto actual y futura.

## 1.1. Contexto y Justificación

En España, el uso de algunas autopistas implica el paso por sistemas de peaje, que a veces pueden ser la causa de congestiones de tráfico, dificultando el flujo de vehículos en la autopista y contribuyendo a la contaminación ambiental. Aunque existan peajes con dispositivos electrónicos que permiten el paso mediante el uso de tarjetas electrónicas, en ocasiones los conductores se confunden, lo que genera atascos de tráfico.

El sistema que se propone a desarrollar en este escenario hipotético se centra en la modificación y eliminación de los dispositivos de peaje actuales, reemplazándolos por un sistema completamente nuevo online. Este nuevo sistema utilizará exclusivamente cámaras de detección de imágenes para facilitar el movimiento de vehículos en la autopista, evitando cualquier dificultad o confusión por parte de los conductores.



**Figura 1.1:** Ejemplo de atasco en sistema de peaje [1]

Es importante destacar que el sistema en desarrollo no representa el sistema real, sino

## 1. VISIÓN GENERAL

---

más bien una versión simplificada debido a la alta complejidad y la falta de recursos para desarrollar el sistema completo.

### 1.2. Público Objetivo

Los principales usuarios y beneficiarios del sistema de peaje son los conductores que utilizan las autopistas con regularidad.

### 1.3. Visión del Proyecto

La visión detrás del sistema de peaje que se está desarrollando es ambiciosa y busca transformar por completo la experiencia de pago de peajes en las autopistas de todo el país. Los objetivos clave que se esperan lograr a través de este sistema incluyen:

1. **Escalabilidad nacional:** El proyecto aspira a crear un sistema de peaje altamente escalable que pueda ser implementado en todas las autopistas del país. Esto permitirá una transición gradual de los sistemas de peaje tradicionales hacia un enfoque moderno y unificado.
2. **Facilidad de mantenimiento:** Se busca diseñar un sistema que sea fácil de mantener y actualizar. Esto garantizará que las futuras expansiones y modificaciones del sistema se realicen de manera eficiente, minimizando los tiempos de inactividad y los costes asociados.
3. **Flexibilidad y adaptabilidad:** El sistema se concibe como una solución flexible y adaptable. Esto significa que podrá evolucionar con las necesidades cambiantes de los conductores y las regulaciones gubernamentales. Se espera que sea capaz de incorporar nuevas tecnologías y métodos de pago a medida que surjan.
4. **Facilitación del paso:** Uno de los objetivos principales es hacer que el paso por los peajes sea lo más fluido y eficiente posible para los conductores. La eliminación de las barreras físicas y la adopción de cámaras de detección de imágenes reducirán los tiempos de espera y las congestiones, mejorando significativamente la experiencia de conducción.
5. **Transparencia y control para los conductores:** El sistema también tiene como objetivo brindar a los conductores una mayor visibilidad y control sobre sus viajes. Esto incluye la capacidad de acceder fácilmente a registros de sus transacciones de peaje y conocer el coste exacto de su viaje, lo que les permitirá tomar decisiones más informadas.

Es importante aclarar que la visión presentada en este capítulo se enfoca principalmente en el sistema real en lugar del sistema simplificado o prototipo. Esto se debe a que el sistema simplificado no incluye funcionalidades físicas como la escalabilidad nacional y la facilitación del paso, debido a limitaciones inherentes.

Es necesario tomar este capítulo con cierta precaución, teniendo en cuenta estas limitaciones. El objetivo principal del sistema simplificado es desarrollar una base que sea

### 1.3. Visión del Proyecto

fácilmente escalable, flexible y mantenable, mientras permite a los usuarios visualizar su información a través de pruebas exhaustivas.

El enfoque actual se centra en garantizar la robustez y la capacidad de adaptación del sistema simplificado, asentando así las bases para un sistema más completo en el futuro. Los detalles específicos sobre el sistema simplificado se presentarán en el siguiente capítulo [2](#).



# 2

## CAPÍTULO

# Planificación

En este capítulo se presentará el sistema de planificación del proyecto, los objetivos, estimaciones de dedicación, cronogramas y los trámites del desarrollo del proyecto.

## 2.1. Alcance del Proyecto

El alcance del proyecto es desarrollar un conjunto de servicios y aplicaciones interrelacionadas para un sistema integral de gestión y cobro vial prototipo. En concreto, se refiere al desarrollo de tecnologías fundamentales: la interfaz de procesamiento de datos y software de visualización de los datos.

### 2.1.1. Descripción del Proyecto Prototipo Virtual

Es importante aclarar que este sistema es un prototipo virtual, diseñado para representar una versión simplificada de un sistema mucho más complejo en su escala real. El propósito de este prototipo es brindar una visión clara y tangible de las funcionalidades y características fundamentales del sistema completo. Cabe mencionar que, a pesar de su carácter simplificado, el prototipo se ha desarrollado con rigurosidad para asegurar que refleje fielmente el comportamiento esperado del sistema real.

### 2.1.2. Objetivos concretos del Proyecto

Los objetivos concretos del proyecto son fundamentales y abarcan aquellos que se encuentran dentro de su alcance. Nos enfocaremos únicamente en estos objetivos concretos para garantizar su correcta implementación.

1. Desarrollar una interfaz que sea capaz de reconocer las matrículas de vehículos y asociarlos con un tramo virtual (en el contexto del sistema un tramo se refiere a una sección específica de una vía o carretera que se delimita entre dos puntos de entrada y salida, que se utiliza para el registro y control de los vehículos que transitan por ese tramo) y guardar esta información en una base de datos relacionado con el usuario del vehículo en cuestión. Se debe especificar que los datos recopilados para las pruebas y

## 2. PLANIFICACIÓN

---

la realización de la demostración permanecerán exclusivamente dentro del ámbito del trabajo de fin de grado. Además, los datos de los usuarios serán ficticios, pero se intentarán asemejar lo más posible a datos reales.

2. Crear una aplicación web que permita el registro y reconocimiento de los usuarios de forma segura. La aplicación debe ser capaz de visualizar los datos de facturación y tramos. La aplicación debe permitir también la modificación, verificación y eliminación de las características necesarias de los usuarios y su información.

### 2.1.3. Objetivos secundarios del Proyecto

Los objetivos secundarios del proyecto abarcan tanto aquellos que pueden ser realizados en un futuro dentro del alcance del proyecto como aquellos que pueden encontrarse fuera de dicho alcance. En nuestro caso, nos enfocaremos únicamente en los objetivos secundarios que estén dentro del alcance del proyecto.

1. El software debe ser capaz de almacenar la localización, hora y tiempo de permanencia en la autopista.
2. En una fase opcional, el software podría reconocer las etiquetas de emisiones CO<sub>2</sub> de los vehículos nacionales y aplicar diferentes normas a los vehículos extranjeros utilizando un servicio específico de la DGT (Dirección General de Tráfico).
3. La aplicación web puede ampliarse a teléfonos móviles.
4. Opcionalmente, se puede incluir estadísticas de los tramos realizados por el usuario, brindándole una visión general de su historial de viajes y ayudándole a gestionar sus desplazamientos de manera más efectiva.
5. Fuera del alcance del proyecto actual, en un futuro, se contempla la posibilidad de desplegar el sistema actual en entornos de producción, lo que permitirá la interacción con usuarios reales.
6. Implementación de algoritmo propio de LPR (License Plate Recognition).

### 2.1.4. Requisitos del Proyecto

Los requisitos del proyecto representan las características que se deben de cumplir para satisfacer los objetivos del proyecto y establecer las funcionalidades, especificaciones y limitaciones del sistema. Los requisitos del proyectos están recogidos dentro del capítulo 3 Captura de Requisitos.

### 2.1.5. Marco Legal

Durante el proceso de desarrollo del sistema en un entorno virtual, se prestará especial atención al marco normativo pertinente para garantizar la adecuación del proyecto. La implementación se ajustará cuidadosamente para asegurar su viabilidad y cumplimiento legal en vistas de una futura utilización. Algunos de los documentos legales que serán considerados en este contexto incluyen:

## **2.2. Descripción de las exclusiones y supuestos**

---

1. Reglamento General de Protección de Datos (RGPD) de la UE[2]: Especifica como garantizar que las organizaciones manejen los datos personales de manera segura y respeten los derechos de privacidad de las personas.
  - Política de Privacidad.
  - Política de Cookies.
  - Términos y Condiciones de Uso.
2. Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPD GDD) de España [3]: Esta ley específica detalla cómo se deben proteger los datos personales en el contexto español y garantiza los derechos digitales de los ciudadanos, como el acceso a la información y la privacidad en línea.

### **2.1.6. Licencia del Proyecto**

La licencia del proyecto que se utilizará será la licencia Creative Commons de Atribución-NoComercial-NoDerivados (CC BY-NC-ND) [4]. Esto significa que se puede utilizar y distribuir el proyecto siempre y cuando se le atribuya como el autor original, no se realicen usos comerciales y no se realicen modificaciones o adaptaciones sin el permiso expreso del autor.

## **2.2. Descripción de las exclusiones y supuestos**

En este apartado se van a identificar y detallar cualquier aspecto o condición que esté explícitamente excluido o no considerado en el alcance del proyecto.

### **2.2.1. Exclusiones**

1. Se excluye la actividad de estudio y exploración de nuevas tecnologías como parte de la dedicación del proyecto. Esta actividad puede llevarse a cabo fuera del horario contemplado para el desarrollo del proyecto.
2. Se excluye cualquier cambio significativo en los requisitos del proyecto una vez que hayan sido definidos y aprobados. Cualquier cambio adicional requerirá una evaluación y aprobación por separado, y no estará dentro del alcance del proyecto actual.
3. Se excluye la actividad de despliegue del sistema en entornos de producción. El sistema se desarrollará y utilizará de forma local, sin considerar la implementación en servidores o infraestructuras de producción.
4. Se excluye la dedicación de tiempo y recursos significativos al análisis extenso de la visión por computadora. Si bien se puede utilizar esta tecnología como parte del reconocimiento de matrículas de vehículos, el enfoque principal del proyecto se centra en el desarrollo de software, así como en la investigación y creación de un sistema escalable y eficiente.

## **2. PLANIFICACIÓN**

---

### **2.2.2. Supuestos**

1. Dado que el proyecto es una idea propia y no hay stakeholders externos o clientes definidos, se asume que los tutores desempeñarán el papel de clientes para proporcionar orientación y retroalimentación sobre el progreso del proyecto. Se supone que los tutores actuarán como representantes del cliente y brindarán la perspectiva y los comentarios necesarios para evaluar el avance y la calidad del trabajo realizado. Su participación será fundamental para validar los requisitos, revisar y aprobar los entregables, y brindar orientación en la toma de decisiones. Los tutores serán considerados como las principales partes interesadas y se espera que colaboren estrechamente con el alumno durante todo el ciclo de vida del proyecto.

### **2.2.3. Entradas al proyecto**

Las entradas al proyecto son todos aquellos elementos, documentos o información que se requieren para iniciar y llevar a cabo el proyecto de manera adecuada.

1. Escenario Teórico/Hipotético: El escenario teórico/hipotético es la idea inicial o visión general del proyecto, que describe el escenario teórico o hipotético en el que se basará el desarrollo. Incluye los objetivos, requisitos y alcance previstos del proyecto.
2. Documento de planificación - plan de ejecución: La documentación de planificación y plan de ejecución proporcionan una descripción detallada de cómo se llevará a cabo el proyecto, incluyendo los pasos específicos, los recursos necesarios, requisitos, objetivos, el cronograma y los entregables esperados. El plan de ejecución es una guía para la implementación del proyecto y se utiliza como referencia durante todas las etapas.

### **2.2.4. Salidas del proyecto**

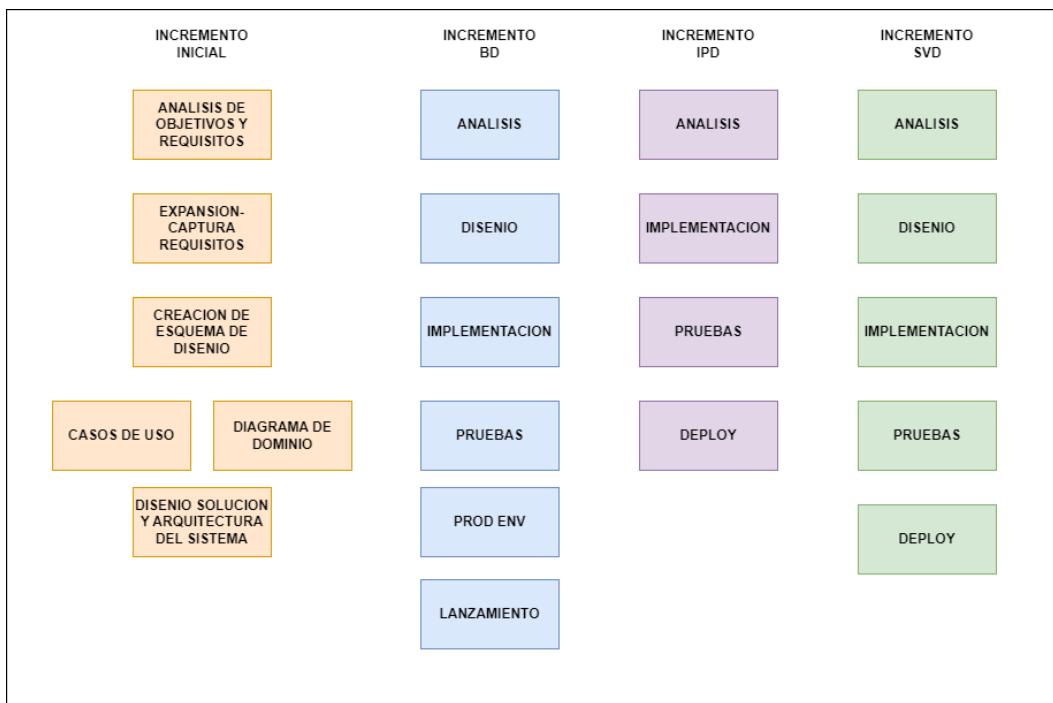
Las salidas del proyecto son los resultados y productos generados como parte del trabajo realizado.

1. Toll sistema o Sistema de peaje: Es el producto principal del proyecto. Consiste en el software desarrollado para realizar el procesamiento de datos, incluyendo el reconocimiento de matrículas de vehículos, almacenamiento de información de tramos, cálculo de facturación, entre otros. También, el software desarrollado para la visualización de los datos recopilados y procesados. Permite a los usuarios acceder y ver la información relevante, como la facturación, tramos recorridos, perfil de usuario, entre otros.
2. Memoria del Trabajo: Documento que recopila toda la información relevante sobre el proyecto, incluyendo los objetivos, requisitos, planificación, metodología utilizada, descripción de los productos desarrollados, resultados obtenidos y conclusiones.

## **2.3. Ciclo de Vida del Proyecto**

El ciclo de vida del proyecto que se ha seleccionado para este proyecto es el iterativo incremental. Esta elección se basa en la necesidad de separar el desarrollo en etapas y

### 2.3. Ciclo de Vida del Proyecto



**Figura 2.1:** Incrementos concretos del ciclo de vida del proyecto

abordar gradualmente los diferentes componentes del sistema. Con este enfoque, podemos centrarnos en el desarrollo del Interfaz de Procesamiento de Datos (IPD) y el Software de Visualización de Datos (SVD) de manera progresiva.

El enfoque iterativo incremental nos permitirá realizar entregas parciales y funcionales del sistema en cada iteración, lo que nos brindará la oportunidad de recibir retroalimentación temprana y realizar ajustes según sea necesario.

Es crucial resaltar que el desarrollo de la base de datos seguirá un enfoque incremental. Comenzaremos con un incremento inicial de la base de datos que nos proporcionará una visión general del sistema de base de datos, lo que facilitará la expansión en cada iteración subsiguiente. Esto implica que la base de datos se construirá de manera gradual, avanzando paso a paso hasta alcanzar su versión final planificada. Esta metodología nos permite centrarnos en las partes específicas del proyecto o funciones que se están desarrollando en ese momento, lo que conduce a un proceso de desarrollo más eficiente y ágil.

- **Incremento Inicial:** Este incremento se basa en el análisis del proyecto entero incluyendo la lectura de la planificación elaborada para la identificación de los requisitos, objetivos y especificaciones sobre el trabajo a desarrollar.
  - **Análisis de objetivos y requisitos:** En esta etapa, se lleva a cabo una revisión exhaustiva de los objetivos del proyecto y los requisitos del sistema. Esto implica identificar y comprender las necesidades y expectativas del cliente, así como los problemas que el software debe resolver.
  - **Expansión de los requisitos y objetivos:** Una vez que se han identificado los objetivos y requisitos iniciales, es necesario expandirlos y detallarlos aún

## 2. PLANIFICACIÓN

---

más. Esto implica definir con mayor precisión lo que se espera del sistema y cómo debe funcionar para satisfacer las necesidades del usuario.

- **Creación de los esquemas de diseño:** En esta etapa, se comienza a diseñar la estructura general del sistema. Esto incluye la creación de diagramas y esquemas que representen cómo se organizará y funcionará el software.
  - **Casos de Uso:** Se elaboran casos de uso que describen las interacciones entre los usuarios y el sistema. Estos casos de uso ayudan a definir las funcionalidades clave del software.
  - **Diagrama de Dominio:** El diagrama de dominio muestra las entidades y sus relaciones en el sistema. Esto ayuda a visualizar la estructura de datos subyacente y cómo se relacionan entre sí.
- **Diseño de la solución y arquitectura del sistema:** Se define la arquitectura general del sistema. Esto incluye la selección de tecnologías y herramientas que se utilizarán en el desarrollo y la creación de un plan de alto nivel para la implementación del software.

### ▪ **Incremento Desarrollo de la Base de Datos (BD):**

- **Análisis:** Definición de los requerimientos de la base de datos y diseño de su estructura inicial.
- **Diseño:** Creación del esquema de la base de datos y definición de las relaciones entre las entidades.
- **Implementación:** Creación de las tablas, índices, vistas y restricciones en la base de datos.
- **Pruebas:** Realización de pruebas sobre el entorno de pruebas para asegurar la integridad y el correcto funcionamiento de la base de datos.
- **Production env:** Entorno de producción para preparar el sistema para lanzamiento.
- **Lanzamiento:** Puesta en marcha de la base de datos cuando se ha verificado su correcta implementación.

### ▪ **Incremento Desarrollo del Interfaz de Procesamiento de Datos (IPD):** Se centra en la selección e implementación del algoritmo de reconocimiento de matrículas, el desarrollo de la aplicación de escritorio y la realización de pruebas para asegurar su correcto funcionamiento.

- **Análisis:** Análisis de las herramientas a utilizar, algoritmos a implementar, datasets, diseño de la interfaz de usuario y funcionalidades de los usuarios y el sistema.
- **Incremento de Base de Datos:** Tras el análisis de las funcionalidades necesarias se creara el incremento IPD de la base de datos.
- **Implementación:** Implementación del código, algoritmos, lógica, y modificaciones para adaptarlo a las necesidades del sistema.
- **Pruebas:** Pruebas de la funcionalidad de los datos, del sistema.

- **Lanzamiento:** Se lanza el IPD cuando se han terminado las pruebas y se ha dado por bueno el funcionamiento del software.
- **Incremento Desarrollo del Software de Visualización de Datos (SVD):**
  - **Análisis:** Identificación de los requisitos de visualización de datos y diseño de la interfaz de usuario.
  - **Incremento de Base de Datos:** Tras el análisis de las funcionalidades necesarias se creara el incremento SVD de la base de datos.
  - **Implementación:** Codificación de la lógica de visualización y creación de la interfaz de usuario.
  - **Diseño:** Elaboración del diseño del sitio web y las funcionalidades estáticas y dinámicas.
  - **Pruebas:** Realización de pruebas para asegurar la correcta visualización de los datos.
  - **Lanzamiento:** Puesta en marcha del software de visualización de datos cuando se ha verificado su correcto funcionamiento.

## 2.4. Estructura de Desglose de Trabajo (EDT)

El EDT, o Estructura de Desglose de Trabajo Figura 2.2, se refiere a la composición de los diferentes paquetes de trabajo que conforman el proyecto.

- Paquete de trabajo EDT TFG: Es el paquete principal que abarca todo el proyecto. Contiene todos los sub-paquetes y representa el nivel más alto de descomposición del proyecto.
  - Paquete de trabajo Análisis y Planificación (AP): Incluye las actividades relacionadas con la identificación de requisitos, la planificación de recursos, análisis de las herramientas y la definición del alcance del proyecto. Aquí se establecen los fundamentos y la dirección del proyecto.
    - Paquete de trabajo Análisis del Dominio (AD): Implica actividades relacionadas con la comprensión y el análisis del sector en la que opera el proyecto.
    - Paquete de trabajo Planificación (P): Se centra en actividades relacionadas con la planificación y gestión del proyecto.
    - Paquete de trabajo Análisis de las Herramientas (AH): Implica la evaluación y análisis de las herramientas, software o tecnologías que se utilizarán en el proyecto.
  - Paquete de trabajo Desarrollo (D): Este paquete abarca todas las etapas del desarrollo del software y los elementos esenciales del sistema, que comprenden el diseño de la interfaz de procesamiento de datos, la gestión de la base de datos y la creación del software de visualización de datos. Para facilitar el proceso de desarrollo del proyecto, se ha dividido en cuatro incrementos fundamentales, cada uno de los cuales desempeña un papel crítico en su evolución. Estos incrementos son:

## 2. PLANIFICACIÓN

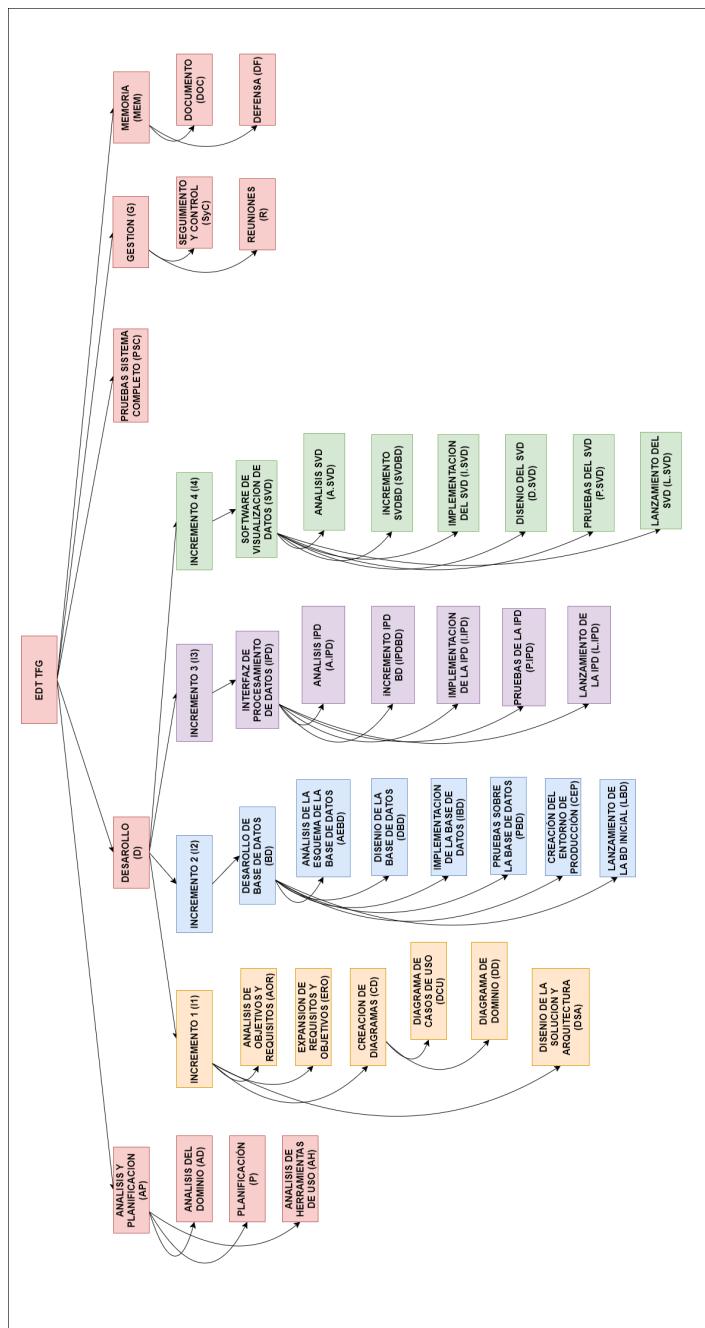


Figura 2.2: Estructura de Desglose de Trabajo

- Incremento 1 (I1): En esta fase, se realiza un análisis exhaustivo de la planificación, los objetivos y los requisitos del proyecto. Su propósito es aclarar y detallar cualquier aspecto pendiente en la propuesta del proyecto. Además, durante esta etapa, se elaboran los diagramas de casos de uso, los diagramas de dominio y se establece la solución y arquitectura del sistema final. Esta etapa desempeña un papel crucial en la definición de una visión clara para el proyecto, fundamental para su éxito.

- Incremento 2 (I2):
  - ◊ Paquete de Trabajo de Base de Datos (BD): Este paquete se concentra en el análisis, diseño, implementación y administración de la base de datos inicial que albergará la información del sistema, abarcando datos como registros de matrículas, perfiles de usuarios y otros datos pertinentes. Además, engloba la creación de entornos tanto de prueba como de producción.
- Incremento 3 (I3):
  - ◊ Paquete de Trabajo de Interfaz de Procesamiento de Datos (IPD): Este paquete se dedica al análisis y desarrollo del software especializado para el procesamiento de datos. Esto involucra la creación de algoritmos para el reconocimiento de matrículas, la definición de métodos de comunicación entre el usuario y el servidor, y la expansión específica de la base de datos para incorporar nuevas relaciones.
- Incremento 4 (I4):
  - ◊ Paquete de Trabajo de Software de Visualización de Datos (SVD): Este paquete se enfoca en el desarrollo del software de visualización de datos, destinado a presentar la información de manera accesible para los usuarios. Esto comprende el diseño de la interfaz de usuario y la implementación de las funcionalidades de visualización.
- Paquete de trabajo Pruebas Sistema Completo (PSC): Engloba las actividades de prueba del sistema completo, incluyendo la integración de los diferentes componentes, las pruebas de funcionalidad, rendimiento y seguridad, así como la corrección de errores y ajustes necesarios.
- Paquete de trabajo Gestión del Proyecto (GP): Este paquete de trabajo está a cargo de la supervisión y control riguroso del proyecto, además de la organización y ejecución de reuniones regulares para garantizar el seguimiento adecuado de su desarrollo.
  - Seguimiento y Control (SyC): Esta parte del paquete de trabajo de Gestión del Proyecto se dedica a asegurarse de que el proyecto avance de acuerdo con el plan establecido. Implica el monitoreo constante de las actividades, la identificación temprana de posibles desviaciones y la implementación de medidas correctivas para mantener el proyecto en el camino correcto.
  - Reuniones (R): Este sub-paquete se enfoca en la organización y realización de reuniones periódicas. Estas reuniones son cruciales para mantener una comunicación efectiva, discutir avances y desafíos, y tomar decisiones para abordar cualquier problema que surja durante la ejecución del proyecto.
- Paquete de trabajo Memoria (MEM): Este paquete tiene la responsabilidad de crear la memoria final del proyecto, que sirve como un registro completo y detallado de los procesos, decisiones, resultados y conclusiones obtenidos a lo largo del desarrollo del proyecto.
  - Documento (DOC): Dentro de este subpaquete, se recopila y organiza toda la información esencial del proyecto de manera sistemática. Esto incluye documentos, informes, gráficos, y cualquier otro material que sea necesario para crear la memoria final. La documentación se estructura de manera lógica y clara, lo que facilita la posterior elaboración de la memoria.

## 2. PLANIFICACIÓN

---

- Defensa (DF): Implica la preparación de material visual, como presentaciones o exposiciones, y la exposición oral de los aspectos clave del proyecto.

### 2.4.1. Periodos de realización de las tareas y fechas de terminación de los entregables

Se refieren a la planificación y programación de las actividades y tareas necesarias para completar el proyecto, junto con las fechas previstas para la finalización de los entregables clave.

#### 2.4.1.1. Descripción tareas a realizar

Tareas a realizar agrupados por paquetes de trabajo:

- Paquete de trabajo Análisis y Planificación (AP):
  - AP.1 Análisis del Dominio (AD)
  - AP.2 Planificación (P)
  - AP.3 Análisis de las Herramientas (AH)
- Paquete de trabajo Desarrollo (D):
  - D.1 Incremento 1 (I1):
    - I1.1 Análisis de Objetivos y Requisitos (AOR)
    - I1.2 Expansión de los Requisitos y Objetivos (ERO)
    - I1.3 Creacion de Diagramas (CD)
      - ◊ CD.1 Diagrama de Casos de Uso (DCU)
      - ◊ CD.2 Diagrama de Dominio (DD)
    - I1.4 Diseño de la Solución y Arquitectura del sistema (DSA)
  - D.2 Incremento 2 (I2):
    - I2.1 Paquete de Trabajo de Base de Datos (BD):
      - ◊ BD.1 Analisis de la Esquema de la Base de Datos (AEBD)
      - ◊ BD.2 Disenio de la Base de Datos (DBD)
      - ◊ BD.3 Implementacion de la Base de Datos (IBD)
      - ◊ BD.4 Pruebas sobre la Base de Datos (PBD)
      - ◊ BD.5 Creacion del Entorno de Produccion (CEP)
      - ◊ BD.6 Lanzamiento de la Base de Datos Inicial (LBD)
  - D.3 Incremento 3 (I3):
    - I3.1 Paquete de Trabajo de Interfaz de Procesamiento de Datos (IPD):
      - ◊ IPD.1 Análisis
      - ◊ IPD.2 Incremento BD
      - ◊ IPD.3 Implementación
      - ◊ IPD.4 Pruebas
      - ◊ IPD.5 Lanzamiento

## 2.4. Estructura de Desglose de Trabajo (EDT)

- D.4 Incremento 4 (I4):
  - I4.1 Paquete de Trabajo de Software de Visualización de Datos (SVD):
    - ◊ SVD.1 Análisis
    - ◊ SVD.2 Incremento BD
    - ◊ SVD.3 Implementación
    - ◊ SVD.4 Diseño
    - ◊ SVD.5 Pruebas
    - ◊ SVD.6 Lanzamiento
- Paquete de trabajo Pruebas Sistema Completo (PSC)
- Paquete de trabajo Gestión del Proyecto (GP):
  - GP.1 Seguimiento y Control (SyC)
  - GP.2 Reuniones (R)
- Paquete de trabajo Memoria (MEM)
  - MEM.1 Documento (DOC)
  - MEM.2 Defensa (DF)

### 2.4.2. Dependencias entre las tareas

Las dependencias entre las tareas son relaciones que indican cómo una tarea afecta o está relacionada con otra dentro del proyecto. Estas relaciones determinan el orden en que deben realizarse las tareas y pueden ser de tipo "predecesora"(una tarea debe completarse antes de que otra comience) o "sucesora"(una tarea depende de la finalización de otra).

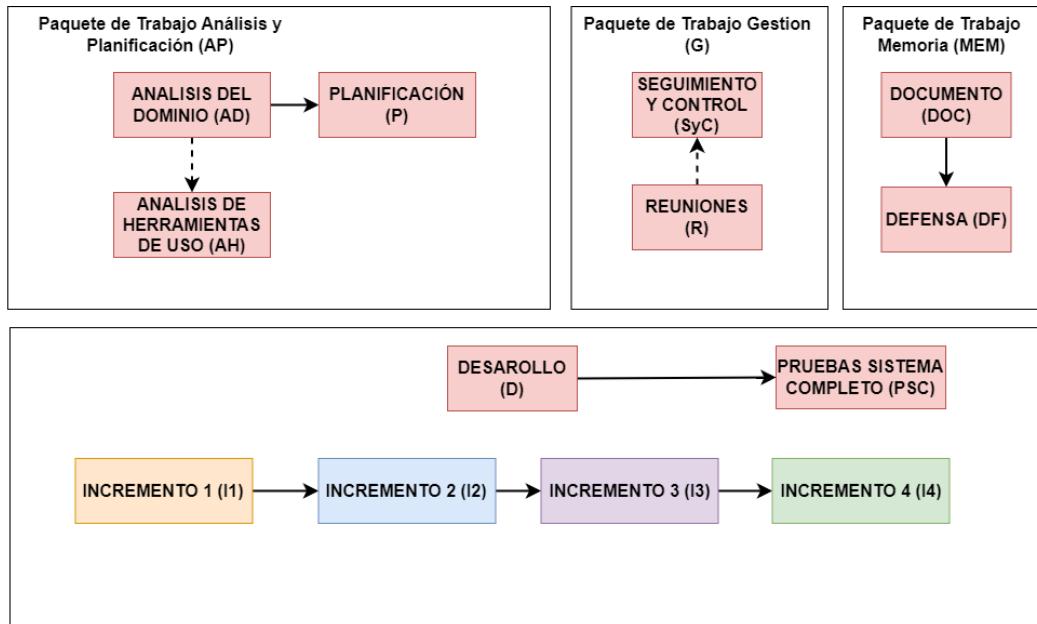


Figura 2.3: Dependencias entre las tareas

## 2. PLANIFICACIÓN

---

### ▪ Dependencias en el Paquete de Trabajo Análisis y Planificación (AP):

- **AP.2 Planificación (P)** depende de **AP.1 Análisis del Dominio (AD)** para comprender los requisitos antes de planificar.
- **AP.3 Análisis de las Herramientas (AH)** puede depender de **AP.1 Análisis del Dominio (AD)** para seleccionar herramientas adecuadas.

### ▪ Dependencias en el Paquete de Trabajo Desarrollo (D):

- **D.1 Incremento 1 (I1)** debe completarse antes de que **D.2 Incremento 2 (I2)** pueda comenzar.
- **D.2 Incremento 2 (I2)** depende de **D.1 Incremento 1 (I1)** para cierta funcionalidad compartida.
- **D.3 Incremento 3 (I3)** depende de **D.2 Incremento 2 (I2)** para acceder a la base de datos.
- **D.4 Incremento 4 (I4)** depende de **D.3 Incremento 3 (I3)** para cierta funcionalidad compartida.

### ▪ Dependencias en el Paquete de Trabajo Pruebas Sistema Completo (PSC):

- Para poder realizar el (PSC) se debe terminar el paquete de Desarrollo (D) antes.

### ▪ Dependencias en el Paquete de Trabajo Gestión del Proyecto (GP):

- **GP.1 Seguimiento y Control (SyC)** puede depender de **GP.2 Reuniones (R)** para obtener información actualizada sobre el progreso del proyecto.

### ▪ Dependencias en el Paquete de Trabajo Memoria (MEM):

- **MEM.2 Defensa (DF)** depende de **MEM.1 Documento (DOC)** para recopilar toda la información necesaria antes de la presentación.

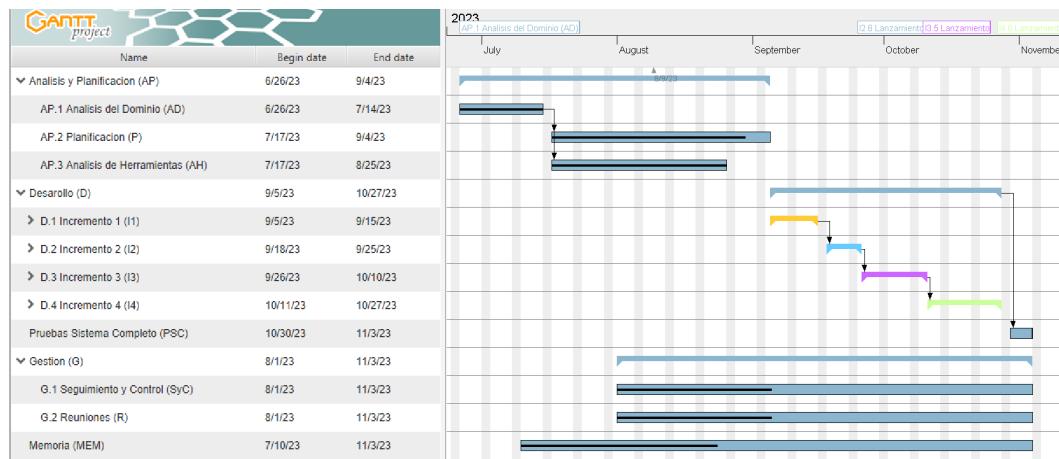
#### 2.4.2.1. Período de desarrollo de las tareas y hitos (Diagrama Gantt)

El diagrama de Gantt, que se presenta en la figura 2.4, es una herramienta esencial en la gestión de proyectos que proporciona una representación visual clara y detallada de la planificación, seguimiento y gestión de las actividades a lo largo del tiempo, lo que ayuda a garantizar el éxito de los proyectos.

En el diagrama diseñado para este proyecto, identificamos cinco hitos cruciales que representan cada uno de los paquetes de trabajo previamente definidos. La planificación y la documentación de memoria son los dos paquetes que requieren una inversión de tiempo más significativa. En primer lugar, la planificación es fundamental, ya que establece la hoja de ruta para la ejecución de las tareas. En segundo lugar, la memoria es un componente crucial, ya que en ella se recopilan y documentan todos los datos generados a lo largo del proyecto.

Además, contamos con el paquete de desarrollo, que abarca la fase de desarrollo desde septiembre hasta finales de octubre. En esta etapa, se espera que la planificación esté completamente finalizada.

## 2.5. Estimación de dedicación a cada una de las tareas



**Figura 2.4:** Diagrama de Gantt

Los dos siguientes hitos son la gestión, que se extiende desde la reunión inicial en julio hasta el cierre final del proyecto, y las pruebas del sistema, destinadas a refinar y perfeccionar el sistema.

## 2.5. Estimación de dedicación a cada una de las tareas

Implica calcular cuántas horas, días o semanas se necesitarán para completar una tarea en particular en función de la complejidad, los recursos disponibles y otros factores relevantes.

Paquete de Trabajo	Tarea	Estimación
Paquete de trabajo Análisis y Planificación (AP)	AP.1 Análisis del Dominio (AD)	4h
	AP.2 Planificación (P)	25h
	AP.3 Análisis de las Herramientas (AH)	5h
	<b>SUBTOTAL AP</b>	<b>34h</b>
Paquete de trabajo Desarrollo-Incremento 1	I1.1 Análisis de Objetivos y Requisitos (AOR)	5h
	I1.2 Expansión de los Requisitos y Objetivos (ERO)	5h
	I1.3 Creación de Diagramas (CD)	10h
	I1.4 Diseño de la Solución y Arquitectura del sistema (DSA)	5h
	<b>SUBTOTAL I1</b>	<b>25h</b>
Paquete de trabajo Desarrollo-Incremento 2-Base de Datos (BD)	BD.1 Análisis de la Esquema de la Base de Datos (AEBD)	3h
	BD.2 Diseño de la Base de Datos (DBD)	10h
	BD.3 Implementación de la Base de Datos (IBD)	15h
		Continúa en la siguiente página

## 2. PLANIFICACIÓN

---

**Tabla 2.1 – Continuación de la página anterior**

Paquete de Trabajo	Tarea	Estimación
Paquete de trabajo Desarrollo- Incremento 3-Interfaz de Procesamiento de Datos (IPD)	BD.4 Pruebas sobre la Base de Datos (PBD)	5h
	BD.5 Creación del Entorno de Producción (CEP)	2h
	BD.6 Lanzamiento de la Base de Datos Inicial (LBD)	1h
	<b>SUBTOTAL BD</b>	<b>36h</b>
Paquete de trabajo Desarrollo- Incremento 3-Interfaz de Procesamiento de Datos (IPD)	IPD.1 Análisis de los algoritmos de uso y selección	5h
	IPD.2 Incremento necesario de la BD	5h
	IPD.3 Implementación de la interfaz.	20h
	IPD.4 Pruebas	15h
	IPD.5 Integración y entrega del IPD.	5h
	<b>SUBTOTAL IPD</b>	<b>50h</b>
Paquete de trabajo Desarrollo- Incremento 4-Software de Visualización de Datos (SVD)	SVD.1 Análisis de los requisitos para el SVD.	5h
	SVD.2 Incremento necesario de la BD	5h
	SVD.3 Implementación del SVD	20h
	SVD.4 Diseño de la aplicación	20h
	SVD.5 Pruebas.	10h
	SVD.6 Integración y entrega del SVD.	2h
	<b>SUBTOTAL SVD</b>	<b>62h</b>
Pruebas (PSC)	<b>SUBTOTAL PSC</b>	<b>20h</b>
Paquete de trabajo Gestión de Proyecto (GP)	GP.1 Seguimiento y Control (SyC)	10h
	GP.2 Reuniones.	8h
	<b>SUBTOTAL GP</b>	<b>18h</b>
Paquete de trabajo Memoria (MEM)	MEM.1 Recopilación de información y datos relevantes	80h
	MEM.2 Preparación para la defensa final del proyecto	7h
	<b>SUBTOTAL MEM</b>	<b>87h</b>
	<b>HORAS TOTALES</b>	<b>332h</b>

**Tabla 2.1:** Tabla de estimación de dedicación de las tareas

---

Se ha asignado la mayor cantidad de horas de trabajo a los dos módulos y paquetes, IPD y SVD, ya que contienen la mayor parte de las tareas en comparación con los demás paquetes. Esto se debe a que incluyen la implementación y las pruebas del sistema, lo cual es fundamental para el éxito del proyecto.

## 2.6. Caracterización del sistema de información y comunicaciones en el proyecto

Esta sección se centra en describir la estructura del proyecto, incluyendo la organización de los archivos y recursos en las diferentes partes del sistema de información y comunicaciones.

### 2.6.1. Estructura del proyecto

- Ficheros del Software de Sistema de Peaje en linea - Toll Sistema:

- Carpeta de código fuente: Contendrá los archivos de código fuente.
- Carpeta de pruebas: Incluirá los archivos de pruebas unitarias y de integración para el sistema.
- Carpeta de recursos: Albergará los recursos adicionales necesarios para el funcionamiento del sistema, como archivos de configuración, data sets adicionales o imágenes.
- Carpeta de documentación: Contendrá la documentación relacionada con el sistema, como manuales de usuario o especificaciones técnicas.

### 2.6.2. Denominaciones

Durante el desarrollo del proyecto se llevará un registro de diferentes recursos de comunicación como reuniones, correos electrónicos, actas y documentos externos. Se establecerán las siguientes denominaciones:

- **Correos Electrónicos:**

- TFG Konstantin: <Titulo>: Esta denominación se utilizará para etiquetar los correos electrónicos relacionados con el Trabajo de Fin de Grado (TFG) de Konstantin. Se incluirá el título específico del correo electrónico después de los dos puntos.
- KA-TFG-23/24: <Titulo>: Esta etiqueta se aplicará a los correos electrónicos que están relacionados con la gestión del proyecto TFG y llevan el número de proyecto KA-TFG-23/24. También se incluirá el título específico del correo después de los dos puntos.

- **Reuniones o Actas de Reuniones:**

- KA-TFG-AR-tt-nn-dd/mm/yyyy: Esta etiqueta se utilizará para nombrar las reuniones o las actas de reuniones relacionadas con el proyecto TFG.
  - “KA-TFG” indica que se trata de un elemento relacionado con el proyecto.
  - “AR” significa “Acta de Reunión”.
  - “tt” representa el número de reunión o acta.
  - “nn” representa el número de versión si es necesario.

## 2. PLANIFICACIÓN

---

- “dd/mm/yyyy” indica la fecha en formato día/mes/año en la que tuvo lugar la reunión o se creó el acta.

### ■ Copias de Seguridad:

Además del sistema de control de versiones se utilizara esta denominación sobre los archivos críticos:

- .bak: Esta extensión de archivo se utilizará para las copias de seguridad del proyecto. Los archivos con esta extensión contendrán una copia de los datos o el código del proyecto en caso de necesidad de restauración.

### ■ Formatos de Archivo:

- .tex Latex, .pdf, .java, .class, .png, .jpeg, .txt, .sql, ...: Estas extensiones de archivo se refieren a los diferentes formatos en los que se pueden encontrar los documentos y recursos relacionados con el proyecto. Por ejemplo, .tex se usa para archivos de LaTeX, .pdf para documentos en formato PDF, .java para archivos de código fuente Java, .png y .jpeg para imágenes, .txt para archivos de texto sin formato, .sql para scripts de bases de datos, entre otros.

### 2.6.3. Copias de Seguridad

Es fundamental garantizar la integridad y disponibilidad de los datos y sistemas utilizados. Una de las mejores prácticas para lograr esto es implementar un sistema de copias de seguridad efectivo. Las copias de seguridad son réplicas de los datos y sistemas originales que se crean con el propósito de restaurarlos en caso de pérdida, daño o corrupción. Como por ejemplo:

- Copias de seguridad del código fuente del proyecto: Se utilizara el sistema de control de versiones y cada incremento significativo o commit se hará una copia de la ultima versión de la carpeta del código para garantizar la seguridad del sistema. Se utilizara un sistema externo como usb u otro dispositivo similar para poder transferir la información en caso necesario. También se utilizara el sistema GitHub<sup>1</sup> para guardar y acceder la información. Se puede decir que un commit significativo se refiere a una entrega o hito relacionado con el incremento del ciclo de vida establecido.
- Copias de seguridad de la memoria: La memoria se exportara en dos formatos (.pdf e .tex) y se utilizara el sistema online Overleaf [5] para asegurar la seguridad del documento.
- Copias de seguridad de la base de datos (datos): Todas las bases de datos deben de tener una copia de seguridad cada día para la base de datos en producción y cada semana para la base de datos en test, en caso de que los datos se pierden, modifiquen o borren, esta copia de seguridad garantizara la recuperación de los datos.

---

<sup>1</sup><https://github.com/>

#### 2.6.4. Comunicaciones

La comunicación en el proyecto se llevará a cabo utilizando diversas herramientas y canales para facilitar la interacción entre el alumno y los tutores. Se utilizarán herramientas de comunicación en línea, como correo electrónico, mensajería instantánea o videoconferencias, para mantener una comunicación fluida y efectiva. Además, se establecerán reuniones, tanto presenciales como virtuales, para discutir el progreso, compartir información y tomar decisiones importantes.

- Correo electrónico: La principal herramienta de comunicación. Se utiliza como método de comunicación de dudas, concretar fechas de reuniones y para el seguimiento y control del proyecto.
- Reuniones presenciales: Las reuniones presenciales se llevarán a cabo en la universidad, lo que permitirá un seguimiento más preciso del proyecto. Al tener a todos presentes físicamente, se facilitará la comunicación y se podrán abordar los aspectos del proyecto de manera más directa y efectiva.
- Reuniones virtuales: Debido a que el alumno no estará en el país a partir del 8 de Julio, las reuniones presenciales serán sustituidas por reuniones virtuales utilizando la herramienta de videoconferencia Microsoft Teams<sup>2</sup>. La utilización de estas herramientas de videoconferencia asegurará que la comunicación y la colaboración continúen de manera efectiva, a pesar de la distancia física. Además, es importante tener en cuenta la diferencia horaria de +1 hora en Bulgaria. Esto significa que al programar las reuniones virtuales, se debe considerar la hora local de los dos países para asegurar la participación de todos los involucrados.

### 2.7. Análisis de los riesgos

Esta sección tiene como propósito analizar y reducir el impacto de los riesgos identificados durante la fase de planificación.

- **R-1 - Riesgos de Tecnologías:**

**Riesgo:** Existe la posibilidad de que surjan problemas o limitaciones debido a la incompatibilidad entre las versiones de las tecnologías utilizadas en el proyecto.

**Nivel (Impacto):** Bajo

**Plan de Prevención:** Para prevenir este riesgo, se llevará a cabo una investigación exhaustiva previa para garantizar la compatibilidad de las versiones de las tecnologías seleccionadas. Se priorizarán aquellas tecnologías que tengan un historial de soporte a largo plazo (LTS) o que se sepa que son compatibles entre sí.

**Plan de Contingencia:** En caso de que surjan incompatibilidades entre las tecnologías utilizadas en el proyecto, se establecerá un plan para la adopción de tecnologías alternativas similares que sean compatibles. Además, se considerará la posibilidad de posibles retrasos de hasta 2 días para realizar las adaptaciones necesarias en las tecnologías y posibles cambios en el proyecto debido a estas incompatibilidades. Este

---

<sup>2</sup><https://teams.microsoft.com>

## 2. PLANIFICACIÓN

---

período adicional de 2 días se destinará a pruebas y ajustes finales para asegurar una transición sin problemas a las tecnologías alternativas.

### ■ R-2 - Riesgo de Calidad de Recursos Externos:

**Riesgo:** Existe la posibilidad de que los recursos externos utilizados en el proyecto, como código de terceros y APIs de terceros, no cumplan con los estándares de calidad requeridos o que se produzcan cambios no previstos en su funcionamiento.

**Nivel (Impacto):** Bajo para recursos externos, Alto para APIs.

**Plan de Prevención:** Para prevenir estos riesgos, se llevará a cabo una investigación exhaustiva y una evaluación rigurosa de los recursos externos y herramientas que se planean utilizar, incluyendo el código de terceros y las APIs de terceros. Además, se documentará detalladamente cómo se utilizan las APIs de terceros en el proyecto. Se mantendrá un seguimiento regular de las condiciones de uso de las APIs y se estará atento a posibles cambios en su funcionamiento.

**Plan de Contingencia:** En caso de que los recursos externos, incluyendo el código de terceros, no cumplan con los estándares de calidad o si las APIs cambian su modelo de pago y esto afecta al proyecto, se considerará la posibilidad de buscar alternativas gratuitas o la creación de una alternativa propia para garantizar la continuidad del proyecto.

### ■ R-3 - Riesgo de Incumplimiento de Plazos:

**Riesgo:** Existe la posibilidad de no poder cumplir con los plazos establecidos para el proyecto.

**Nivel (Impacto):** Medio

**Plan de Prevención:** Elaboración de un cronograma realista y seguimiento periódico del progreso del proyecto.

**Plan de Contingencia:** Preparar un plan de ajuste del cronograma y asignación adicional de recursos en caso de retrasos de una o dos semanas.

### ■ R-4 - Riesgo de Datos Insuficientes o Erróneos para Pruebas (Reconocimiento de Matrículas):

**Riesgo:** Existe la posibilidad de que los datos utilizados para las pruebas, en particular para el reconocimiento de matrículas, sean insuficientes o contengan errores.

**Nivel (Impacto):** Medio

**Plan de Prevención:** Para prevenir este riesgo, se establecerá un proceso de pruebas riguroso que incluirá la verificación de la calidad de los datos utilizados. Se obtendrán datos de forma manual recogiendo imágenes de matrículas en la calle para garantizar resultados más realistas. Además, se explorarán opciones de búsqueda de datasets de matrículas disponibles que puedan ser utilizados en las pruebas.

**Plan de Contingencia:** En caso de que los datos disponibles resulten insuficientes o defectuosos, se preparará un plan de contingencia que incluirá la modificación de conjuntos de datos existentes o la búsqueda de alternativas adecuadas para asegurar que las pruebas puedan llevarse a cabo de manera efectiva.

■ **R-5 - Riesgos de Empleabilidad:**

**Riesgo:** Posibilidad de encontrar trabajo antes de completar el proyecto.

**Nivel (Impacto):** Medio

**Plan de Prevención:** No aplicable.

**Plan de Contingencia:** Reunión con tutores para replanificar el proyecto y ajustar los hitos si es necesario.

■ **R-6 - Riesgo de Comunicación Ineficiente con Tutores:**

**Riesgo:** Existe la posibilidad de que la comunicación con los tutores del proyecto sea insuficiente, dado que son los principales interesados en el proyecto. Dado que el proyecto es una iniciativa propia, los tutores y el alumnado son los principales stakeholders.

**Nivel (Impacto):** Bajo

**Plan de Prevención:** Para prevenir este riesgo, se llevará a cabo una comunicación regular y efectiva con los tutores. Se programarán reuniones periódicas cada 15 días y se utilizarán correos electrónicos si fuera necesario entre ellas para mantener a los tutores informados sobre el progreso del proyecto y cualquier cambio relevante.

**Plan de Contingencia:** En caso de que la comunicación con los tutores se vuelva deficiente, se establecerá un plan de contingencia que incluirá la programación de reuniones adicionales, la documentación de cualquier cambio importante y la implementación de una comunicación efectiva mediante la compartición de recursos si es necesario.



# 3

## CAPÍTULO

# Captura de Requisitos

En este capítulo, se ampliarán los requisitos que se definieron inicialmente después de analizar la propuesta del proyecto. Esta expansión proporcionará una visión más clara del desarrollo de los componentes y simplificará su implementación.

## 3.1. Stakeholders y Usuarios

Esta sección tiene como propósito brindar claridad a los stakeholders que participan en el proyecto y llevar a cabo un análisis exhaustivo de las necesidades y expectativas de los usuarios finales.

### 3.1.1. Identificación de Stakeholders

Lista de las partes interesadas en el proyecto, incluyendo clientes, usuarios finales y otros actores clave.

Como se estableció al inicio del proyecto, este es una propuesta original del alumno que se fundamenta en un escenario hipotético. En esta línea, los posibles clientes hipotéticos del proyecto podrían incluir a la **Dirección General de Trafico**, con el objetivo de ampliar la utilización de los sistemas de peaje existentes.

Los usuarios finales son **aquellos individuos que conducen o son propietarios de vehículos, y utilizan el sistema de peaje** como método de acceso a las carreteras que requieren su uso.

Como se ha establecido previamente, este proyecto se centra en una versión simplificada del sistema real, lo que implica que no se definirán recursos adicionales ni otros actores importantes en el desarrollo del sistema, ya que no forman parte del alcance de este proyecto.

### 3.1.2. Identificación de los actores del sistema

Los actores que interactuarán con el sistema se dividen en dos roles principales: el usuario autenticado y el usuario no autenticado. Estos dos roles utilizarán el Sistema de Visualización de Datos (SVD) para interactuar con la plataforma. Además de estos roles,

### **3. CAPTURA DE REQUISITOS**

---

existen otros actores adicionales, como el administrador y el personal, que no están incluidos en el alcance actual del proyecto.

Adicionalmente, existe un actor especial dentro del sistema que desempeña el papel del propio sistema. Este rol no se refiere específicamente a un usuario real, sino que representa la funcionalidad interna del sistema. Este actor se ha incluido para facilitar la simulación y demostración de las numerosas funcionalidades automáticas que el sistema posee internamente. Esta inclusión es necesaria para poder presentar de manera efectiva las capacidades del sistema en el contexto del proyecto actual.

#### **3.1.3. Requisitos Funcionales**

Los requisitos funcionales describen las funciones que el sistema debe de cumplir para cumplir los objetivos del proyecto.

##### **1. Interfaz de procesamiento de datos:**

- Reconocimiento de matrículas de vehículos: Permitirá la identificación automática de las matrículas de los vehículos que ingresan al tramo y registrar la información necesaria en la base de datos.
- Almacenamiento de localización, hora y tiempo de permanencia de los vehículos en un tramo: Opcional para mantener un registro de los vehículos y sus actividades en el tramo. La información de la localización permitirá saber los dos puntos de entrada y salida del tramo, mientras la hora y tiempo de permanencia proporcionarán datos relevantes para el cálculo de la facturación.
- Selección de un tramo. Este requisito permitirá seleccionar el tramo específico que se desea testear para ciertos usuarios (Esto no pasará en un sistema real, pero como los tramos son virtuales se pueden seleccionar para ciertos datos de testeo para simular el comportamiento del sistema real).
- Comunicación con base de datos: Esta interfaz permitirá el intercambio de información entre el software y la base de datos de forma estandarizada y segura.
- Cálculo de facturación/cobro dependiendo de las características establecidas: Este requisito opcional, implica el desarrollo del sistema de cálculo de facturación basada en las características establecidas.

##### **2. Software de visualización de datos:**

- Autenticación y Registro de usuarios: Este requisito implica la creación de un sistema de registro, inicio de sesión, a través de un formulario con los datos necesarios (nombre, cuenta bancaria/tarjeta, correo, contraseña, vehículo de usuario).
- Visualización de facturación: Proporciona a los usuarios su información de facturación, como los importes cobrados por el uso de los tramos y las transacciones realizadas.
- Visualización de tramos y características concretas con el tramo: Permite a los usuarios visualizar la información de los tramos, incluyendo aquellos por los que han transitado y la respectiva facturación asociada a cada tramo.

- Visualización perfil usuario y gestión: Este requisito se refiere a la capacidad de los usuarios para ver y gestionar su perfil personal en el sistema.

### 3.1.4. Requisitos No Funcionales

Los requisitos no funcionales se refieren a las características y restricciones del sistema que no están relacionadas directamente con sus funcionalidades.

#### Rendimiento

##### 1. Optimización del procesamiento de datos:

- **Métrica:** Tiempo promedio de procesamiento de datos por transacción.
- **Cómo medirlo:** Registra el tiempo necesario para procesar una transacción de peaje y calcula el tiempo promedio.
- **Objetivo:** Menos de 2500 milisegundos (2.5 segundos) por transacción.

#### Mantenibilidad

##### 1. Facilidad de mantenimiento:

- **Métrica:** Tiempo promedio de incorporación de nuevas funcionalidades o cambios.
- **Cómo medirlo:** Registrar el tiempo que lleva incorporar nuevas funcionalidades o realizar cambios en el sistema. Calcula el tiempo promedio de incorporación a lo largo de varias modificaciones o adiciones.
- **Objetivo:** Un promedio de menos de una semana por cada adición o cambio importante.

##### 2. Documentación:

- **Métrica:** Cobertura de documentación.
- **Cómo medirlo:** Evaluar la cantidad de código, arquitectura y documentación de diseño disponible.
- **Objetivo:** Al menos el 80 % del código esté documentado.

#### Tolerancia a Fallos

##### 1. Tasa de fallos en el reconocimiento de matrícula:

- **Métrica:** Número de fallos en el reconocimiento de matrícula por cada 10 intentos.
- **Cómo medirlo:** Registrar el número de fallos en el reconocimiento de matrícula en un período determinado y calcula la tasa de fallos por cada 10 intentos.
- **Objetivo:** Menos de 1 fallo por cada 10 intentos.

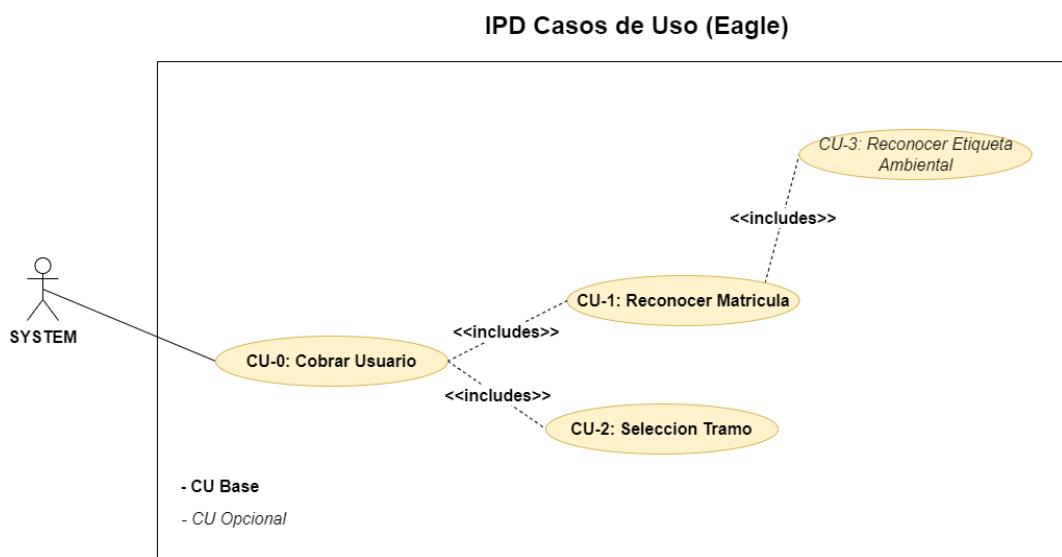
### 3. CAPTURA DE REQUISITOS

#### 3.2. Casos de Uso

En esta sección, se detallan los casos de uso del sistema

##### 3.2.1. Diagrama de Casos de Uso del modulo IPD

Se presenta el diagrama de casos de uso del módulo del sistema IPD. Es importante destacar que tanto el sistema actual como el diagrama de casos de uso representan una versión simplificada y no el sistema real. El sistema real incluye una cantidad significativamente mayor de casos de uso que no están contemplados en el sistema actual.



**Figura 3.1:** Diagrama de casos de uso IPD

A continuación se explica cada caso de uso de la figura 3.1:

- **CU-0: Cobrar Usuario** Este caso de uso constituye el núcleo del sistema de cobro, ya que, una vez que el sistema ha detectado que un vehículo ha transitado por un tramo específico, procede automáticamente a realizar el cobro correspondiente. Esta automatización es esencial para mantener la fluidez del sistema, evitando la necesidad de que los usuarios realicen pagos manuales a través de la aplicación u otros medios.
- **CU-1: Reconocer Matricula** Este caso de uso se refiere al proceso en el que el sistema de peaje captura y reconoce automáticamente la matrícula del vehículo que se acerca al punto de peaje. El sistema utiliza cámaras de reconocimiento para obtener esta información. En la versión simplificada del sistema actual, se prescindirá del uso de cámaras y, en su lugar, se empleará un sistema de administración con imágenes preseleccionadas para realizar pruebas y evaluar el funcionamiento del sistema.
- **CU-2: Selección Tramo** Cada tramo representa una ruta en la carretera que conecta dos puntos: uno de entrada y otro de salida. El reconocimiento de matrícula se lleva a cabo en los puntos de entrada y salida para identificar las matrículas y luego asociarlas al usuario del vehículo correspondiente. Una vez que se ha completado la captura de datos del usuario del tramo, el usuario puede conocer el costo actual o proyectado

del tramo por el que ha transitado, ya que se le mostrará la información necesaria. Es importante destacar que la selección del tramo se realizará exclusivamente a través del sistema de administración, ya que no es posible llevar a cabo pruebas adicionales más cercanas al sistema real.

- **CU-3: Reconocer Etiqueta Ambiental** Este caso de uso permite al sistema proporcionar un cobro más completo, dependiendo de si se alcanza el objetivo relacionado con la etiqueta ambiental.

### 3.2.1.1. Diagrama de Casos de Uso SVD

Se presenta el diagrama de casos de uso del módulo del sistema SVD (Sistema de Visualización de Datos).

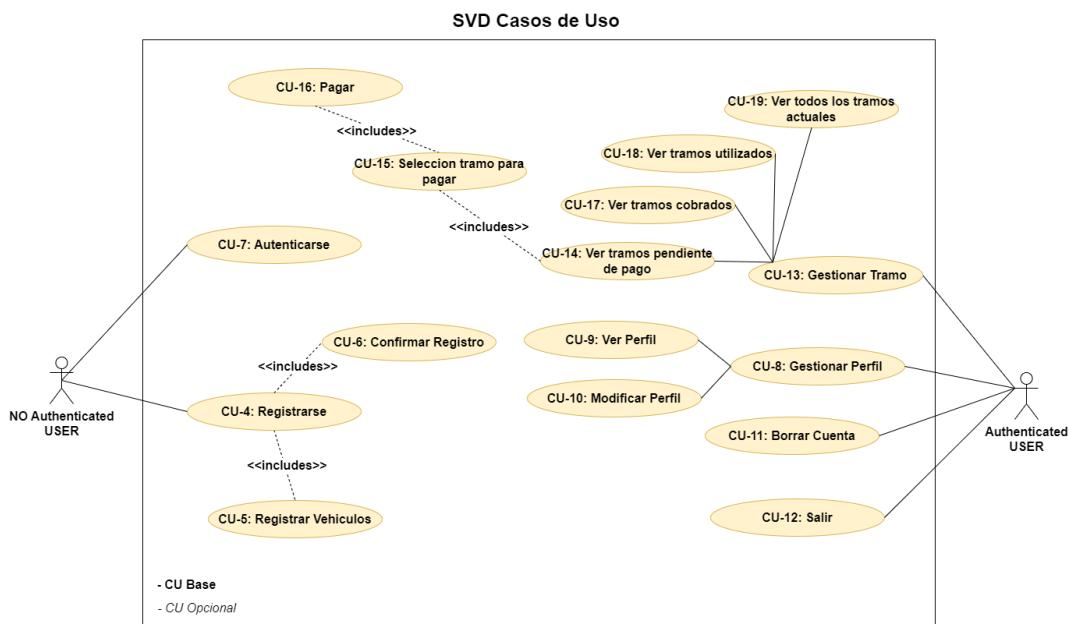


Figura 3.2: Diagrama de casos de uso SVD

A continuación se explica cada caso de uso de la figura 3.2:

- **CU-4: Registrarse** Este caso de uso permite a los usuarios crear una cuenta en el sistema proporcionando la información requerida para la identificación y autenticación en futuros accesos.
- **CU-5: Registrar vehículo** Con este caso de uso, los usuarios pueden ingresar información sobre sus vehículos, como el modelo, la marca y la placa, para que queden registrados en el sistema.
- **CU-6: Confirmar registro** Despues de registrarse, los usuarios pueden utilizar este caso de uso para confirmar su registro y activar su cuenta, generalmente mediante un enlace de confirmación enviado a su correo electrónico.

### 3. CAPTURA DE REQUISITOS

---

- **CU-7: Autenticarse** Los usuarios pueden utilizar este caso de uso para iniciar sesión en el sistema, proporcionando sus credenciales (nombre de usuario y contraseña) para verificar su identidad.
- **CU-8: Gestionar perfil** Este caso de uso permite a los usuarios administrar su información de perfil, que puede incluir datos como nombre, dirección, número de teléfono, entre otros.
- **CU-9: Ver perfil** Los usuarios pueden usar este caso de uso para visualizar su perfil y revisar la información almacenada en él.
- **CU-10: Modificar Perfil** Con este caso de uso, los usuarios pueden realizar cambios en su perfil, como actualizar información personal o cambiar la contraseña.
- **CU-11: Borrar Cuenta** Los usuarios tienen la opción de eliminar permanentemente su cuenta a través de este caso de uso, lo que resultará en la eliminación de toda su información y registros asociados.
- **CU-12: Salir** Este caso de uso permite a los usuarios cerrar sesión de manera segura en el sistema, evitando el acceso no autorizado.
- **CU-13: Gestionar tramos** Los usuarios tienen la capacidad de visualizar los tramos por los que han transitado, ubicarlos en un mapa para obtener una referencia geográfica y revisar los tramos que han sido cobrados, así como aquellos que están pendientes de pago. Con la implementación de este caso de uso, se concluyen los casos de uso principales y de mayor relevancia del sistema.
- **CU-14: Ver tramos pendientes de pago** Permite a los usuarios acceder a una lista de tramos que aún no han sido pagados y que requieren atención por parte del usuario. El sistema ofrecerá a los usuarios la opción de realizar los pagos de manera automática o manualmente desde la aplicación web.
- **CU-15: Selección tramo para pagar** Los usuarios pueden utilizar este caso de uso para elegir un tramo específico de la lista de pendientes de pago y proceder al proceso de pago.
- **CU-16: Pagar** El sistema simplificado actual incorporará además la opción de pago manual con el propósito de llevar a cabo pruebas exhaustivas del flujo de trabajo del sistema.
- **CU-17: Ver tramos cobrados** Los usuarios pueden consultar una lista de tramos que ya han sido cobrados con éxito.
- **CU-18: Ver tramos utilizados** Este caso de uso permite a los usuarios revisar una lista de los tramos que han utilizado, independientemente de si han sido cobrados o no.
- **CU-19: Ver todos los tramos actuales** Con este caso de uso, los usuarios pueden acceder a una lista completa de todos los tramos disponibles en el sistema en ese momento.

### 3.3. Diagrama de Dominio

El diagrama de dominio 3.3 representa las entidades y sus relaciones en el dominio del problema.

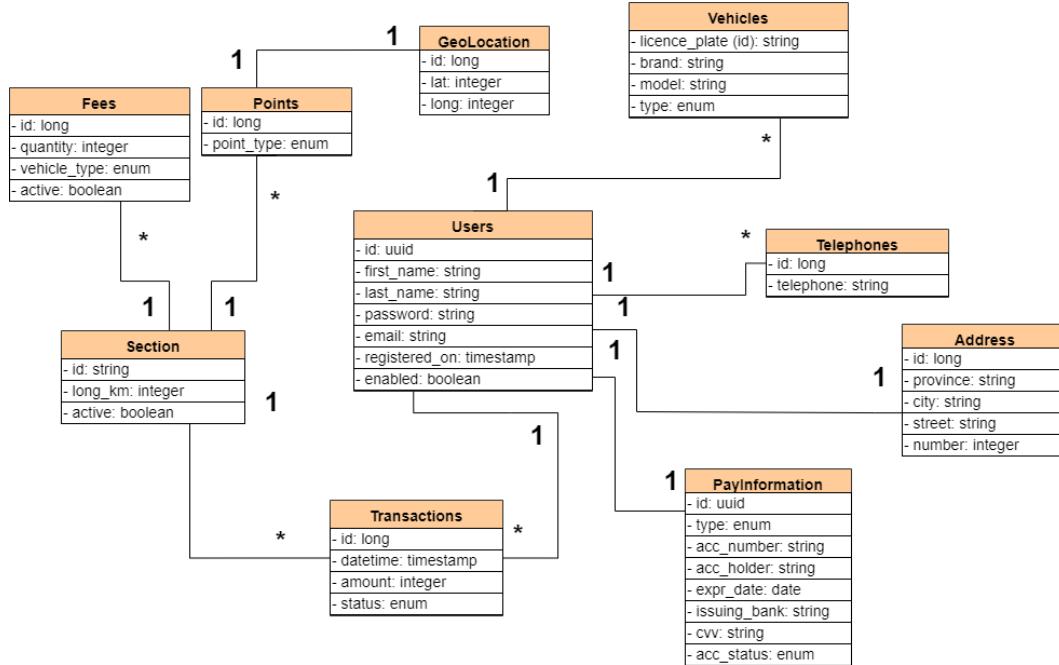


Figura 3.3: Diagrama de Dominio

Aquí están todas las entidades que se han identificado en el proyecto.

#### 3.3.1. Entidades del Dominio

Las principales entidades que están involucradas en el sistema y sus atributos relevantes son:

- **Fee**: Almacena información sobre las tarifas de peaje aplicadas en diferentes tramos o momentos.
  - **id**: Un identificador único que permite distinguir cada tarifa de las demás, de tipo long (auto generado).
  - **quantity**: La cantidad en euros que se cobra como tarifa de peaje para un tramo específico, en decimal.
  - **vehicle\_type**: Indica el tipo de vehículo al que se aplica esta tarifa (por ejemplo, automóvil, camión, motocicleta, etc.), es un enum.
  - **activo**: Si esta activa esta tarifa para el tramo, un booleano.
- **Point**: Ubicaciones específicas en los tramos o puntos de entrada/salida en la autopista.
  - **id**: Identificador único para el punto, long (auto generado).

### 3. CAPTURA DE REQUISITOS

---

- **point\_type:** Si es de entrada o salida, enum.
- **Section:** Representa segmentos o secciones de la autopista o carretera donde se pueden aplicar peajes o donde ocurren eventos relevantes.
  - **id:** Este atributo sirve como identificador único del tramo y puede tomar la forma de un número de identificación o el número de la autopista, por ejemplo, (AP-8), es un string entre 2 y 20 caracteres.
  - **long\_km:** Longitud del tramo en kilómetros, en decimal.
  - **activo:** Si está activo el tramo, un booleano.
- **User:** Esta entidad representa a las personas que utilizan la autopista y que pueden interactuar con el sistema de peaje.
  - **id:** Identificador único del usuario, un uuid.
  - **first\_name:** Primer nombre del usuario, string.
  - **last\_name:** Apellido del usuario, string.
  - **password:** Contraseña del usuario, string entre 8 y 18 caracteres.
  - **email:** Correo electrónico del usuario, string.
  - **registered\_on:** Fecha y hora de registro del usuario, date.
  - **enabled:** Estado de activación del usuario (habilitado o deshabilitado), boolean.
- **Vehicle:** Representa los vehículos que utilizan la autopista y están sujetos al sistema de peaje. Incluye información sobre el tipo de vehículo, su propietario y otros detalles relevantes.
  - **license\_plate:** Matrícula del vehículo, string entre 4 y 20 caracteres.
  - **brand:** Marca del vehículo, un string.
  - **model:** Modelo del vehículo, string.
  - **type:** Tipo de vehículo, enum.
- **Transaction:** Registra las transacciones de peaje, incluyendo información sobre cuándo, dónde y cuánto se pagó.
  - **id:** Identificador único de la transacción, long (auto incrementado).
  - **datetime:** Fecha y hora de la transacción, timestamp.
  - **amount:** Monto o cantidad de la transacción, decimal.
  - **status:** Estado de la transacción (por ejemplo, completada, pendiente, anulada, etc.) enum.
- **Telephone:** Representa las roles de autoridad de los usuarios.
  - **id:** Identificador único del número de teléfono, long (auto generado).
  - **telephone:** Número de teléfono del usuario, string.
- **Address:** Representa las roles de autoridad de los usuarios.

- **id:** Identificador único de la dirección, long (auto generado).
  - **province:** Provincia de la dirección, string.
  - **city:** Ciudad de la dirección, string.
  - **street:** Nombre de la calle de la dirección, string.
  - **number:** Número de la calle de la dirección, integer.
- **PayInformation:** Representa las roles de autoridad de los usuarios.
- **id:** Identificador único de la información bancaria, uuid (auto generado).
  - **type:** Tipo de cuenta bancaria (por ejemplo, cuenta de ahorros, cuenta corriente, tarjeta de crédito, etc.), enum.
  - **acc\_number:** Número de cuenta bancaria, string.
  - **acc\_holder:** Nombre del titular de la cuenta bancaria, string.
  - **expr\_date:** Fecha de vencimiento de la cuenta o tarjeta (si es aplicable), date.
  - **issuing\_bank:** Banco emisor de la cuenta o tarjeta, string.
  - **cvv:** Código de seguridad de la cuenta o tarjeta (si es aplicable), string.
  - **acc\_status:** Estado de la cuenta bancaria (por ejemplo, activa, inactiva, bloqueada, etc.), enum.
- **GeoLocation:**
- **id:** Identificador único de la localización geográfica, long (auto generado).
  - **lat:** Latitud, decimal.
  - **long:** Longitud, decimal.

### 3.3.2. Relaciones

En esta sección se describen las relaciones entre las entidades definidas arriba y como se relacionan con las funcionalidades del sistema.

#### 3.3.2.1. Section, Point, Fee y GeoLocation

El tramo es la entidad principal en la parte del IPD (Interfaz de Procesamiento de Datos) y es fundamental para el funcionamiento del sistema. Como se ha definido previamente, un tramo se utiliza para identificar una sección de la carretera, delimitada por varios puntos de entrada y salida, cada uno de los cuales está asociado con una ubicación geográfica específica para proporcionar información geográfica precisa. Además, cada tramo tiene sus propias tarifas para diferentes tipos de vehículos.

Con esta información, cuando se realiza un reconocimiento de matrícula, se puede establecer una relación entre la matrícula y un usuario donde se guarda en Transaction.

#### 3.3.2.2. Transaction

Las transacciones representan un historial de los tramos que los usuarios han transitado.

### 3. CAPTURA DE REQUISITOS

#### 3.3.2.3. Users, Vehicle, PayInformation, Telephone y Address

Los usuarios deben proporcionar información básica, números de teléfono móvil, dirección, información bancaria y detalles sobre el vehículo que conducen. Todos estos datos son obligatorios para que un usuario pueda utilizar el sistema.

## 3.4. Diseño de la Base de Datos

Esta sección se centra en el diseño de la base de datos que respalda el sistema. A diferencia del diagrama de dominio, proporciona información más detallada, ya que se analiza detalladamente el dominio del problema. Además, incorpora modificaciones resultantes del proceso de diseño y normalización de la base de datos.

### 3.4.1. Modelo de Datos

Se presenta el modelo de la base de datos de diseño correspondiente en la figura 3.4.

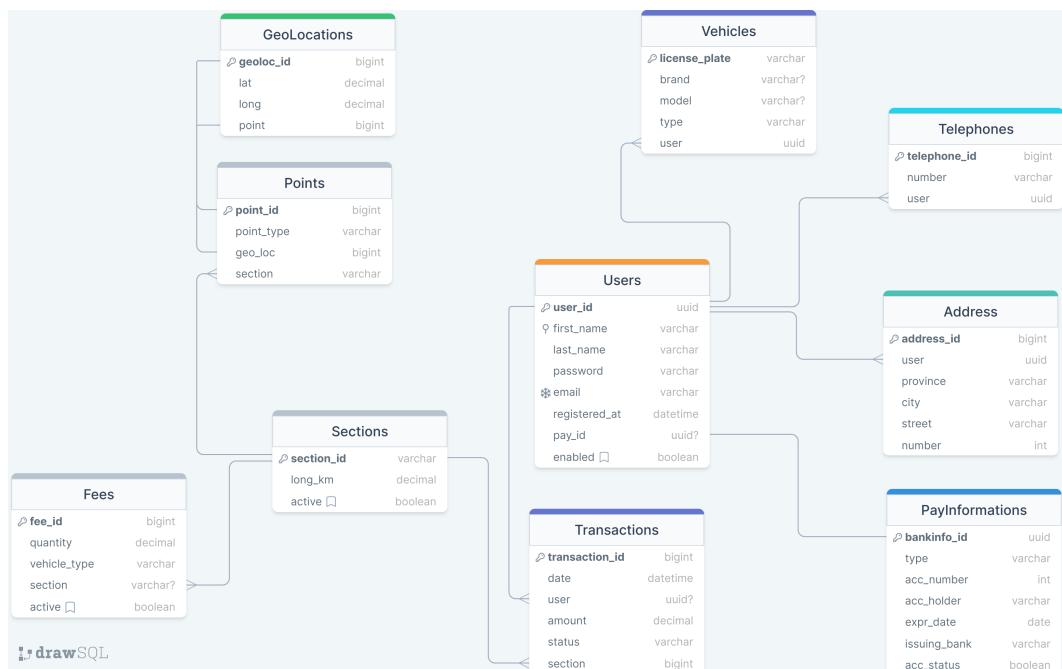


Figura 3.4: Diagrama de Diseño de la Base de Datos

### 3.4.2. Normalización

Dentro del proceso de diseño de la base de datos, uno de los aspectos fundamentales es la aplicación de la normalización para mantener la integridad de los datos [6]. En este proceso, hemos llevado a cabo la normalización hasta alcanzar la Tercera Forma Normal (3FN), que representa un nivel avanzado en la estructuración de la base de datos.

- **Cumplimiento de la Primera Forma Normal (1FN):** Cada tabla debe tener una clave primaria única y las columnas deben contener valores atómicos, es decir, no deben ser listas o conjuntos.

- **Cumplimiento de la Segunda Forma Normal (2FN):** Cada columna no clave debe depender completamente de la clave primaria. No debe haber dependencias parciales de las columnas no clave con respecto a la clave primaria.
- **Cumplimiento de la Tercera Forma Normal (3FN):** No debe haber dependencias transitivas entre las columnas no clave y la clave primaria. Esto significa que si una columna no clave depende de otra columna no clave, esa relación debe ser eliminada.



# 4

## CAPÍTULO

# Herramientas y Tecnologías

En este capítulo, se llevará a cabo un análisis detallado de las tecnologías y herramientas que se emplearán en el desarrollo del proyecto, incluyendo una comparativa con otras opciones y análisis de sus ventajas y desventajas.

## 4.1. Herramientas

En esta sección, se detallarán las herramientas que se emplearán tanto en el desarrollo del proyecto como en las pruebas correspondientes.

### 4.1.1. IntelliJ IDEA

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) de JetBrains diseñado específicamente para Java. Se utilizará para la codificación y el desarrollo de la aplicación principal del proyecto, aprovechando su potente conjunto de herramientas para la programación en Java.

### 4.1.2. IntelliJ DataGrip

IntelliJ DataGrip es otra herramienta de JetBrains, especializada en la gestión de bases de datos. Será fundamental para diseñar y administrar la base de datos que respalda la aplicación, lo que incluye la creación de tablas, consultas y la gestión general de la base de datos.

### 4.1.3. IntelliJ WebStorm

Este IDE de JetBrains está orientado al desarrollo web, lo que lo hace ideal para la creación y el mantenimiento de la interfaz de usuario basada en React y JavaScript. Facilitará el diseño y la implementación de la parte front-end de la aplicación.

## 4. HERRAMIENTAS Y TECNOLOGÍAS

---

### 4.1.4. VSCode

Visual Studio Code (VSCode) es un editor de código altamente personalizable y ampliamente utilizado. Se empleará para tareas relacionadas con el desarrollo web y otros aspectos del proyecto que requieran una herramienta más ligera y flexible que un IDE completo.

### 4.1.5. MySQL Workbench

MySQL Workbench es una herramienta de administración de bases de datos que se utilizará en conjunto con IntelliJ DataGrip para interactuar con la base de datos MySQL del proyecto. Facilitará la modelización de datos, la optimización de consultas y la gestión general de la base de datos.

## 4.2. Aplicación Web

En esta sección, se llevará a cabo un análisis exhaustivo en el desarrollo de una aplicación web, centrándose en diversos aspectos clave:

### 4.2.1. Experiencia del usuario en las aplicaciones web

El sistema requiere el uso de recursos online, especialmente debido al uso de la API externa que se necesita para el procesamiento de los datos. Esto significa que los usuarios deben mantener una conexión a Internet activa para utilizar el sistema de manera efectiva. [7]

Ventajas	Desventajas
En la aplicación web, los usuarios pueden acceder al servicio de forma sencilla a través de una URL sin necesidad de instalar nada en su dispositivo	La falta de conexión a Internet limita la funcionalidad de las aplicaciones, ya que no se pueden acceder ni utilizar sin una conexión en línea

Tabla 4.1: Ventajas y desventajas en experiencia del usuario

### 4.2.2. Consideraciones de rendimiento en términos de velocidad y recursos

Las aplicaciones web ofrecen una mayor compatibilidad y accesibilidad, ya que se ejecutan en navegadores web estándar en una variedad de dispositivos y sistemas operativos

### 4.2.3. Aspectos de seguridad

En esta sección, se analizarán los aspectos de seguridad que son de particular importancia para los tipos de aplicaciones que estamos considerando en este proyecto:

- **Autenticación y Autorización:** La autenticación y autorización se apoyan en roles para garantizar que solo los usuarios autorizados tengan acceso y puedan realizar acciones específicas, como efectuar el pago de peajes.

- **Seguridad de la Comunicación:** En el contexto actual, se utiliza HTTP para la comunicación entre el cliente y el servidor. Sin embargo, es importante destacar que, en un entorno de producción expandido, se tiene la intención de implementar HTTPS como medida de seguridad adicional. HTTPS garantiza la protección de los datos transmitidos, previniendo la interceptación y manipulación de la información durante la comunicación. Este enfoque se adoptará para fortalecer la seguridad de la comunicación en un futuro despliegue en producción.
- **Prevención de Ataques Web:** Estas medidas se centran en proteger la aplicación web contra dos tipos de ataques comunes. El Cross-Site Scripting (XSS) busca evitar que los atacantes inserten scripts maliciosos en la aplicación, mientras que el Cross-Site Request Forgery (CSRF) protege contra ataques en los que un usuario es engañado para realizar acciones no deseadas en su cuenta. Estas medidas son esenciales para mantener la seguridad de la aplicación web y la privacidad de los usuarios. [8]
- **Gestión de Sesiones y Cookies:** Las sesiones permiten mantener la información del usuario durante su interacción con la aplicación, mientras que las cookies son pequeños archivos almacenados en el navegador del usuario que almacenan información relevante.

#### 4.2.4. Selección del enfoque más adecuado para satisfacer los objetivos del proyecto.

En el contexto del desarrollo del sistema de peaje, la elección de una aplicación web se muestra como la opción óptima debido a la necesidad de implementar múltiples servicios específicos. Además, se considera factible la futura expansión de esta aplicación web hacia una aplicación móvil.

### 4.3. REST vs SOAP

REST (Representational State Transfer) y SOAP (Simple Object Access Protocol) son dos protocolos utilizados para la comunicación entre aplicaciones.

#### 4.3.1. Comparación de REST y SOAP

A continuación se presenta la comparación de ambos:

##### 4.3.1.1. Protocolo y Estilo de Arquitectura

- REST es un estilo de arquitectura basado en HTTP. Utiliza los métodos HTTP como GET, POST, PUT y DELETE para operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en recursos.
- SOAP es un protocolo de comunicación que puede utilizar diferentes protocolos subyacentes, como HTTP, SMTP o TCP. Está basado en XML.

## **4. HERRAMIENTAS Y TECNOLOGÍAS**

---

### **4.3.1.2. Formato de Datos**

- REST generalmente utiliza formatos de datos livianos como JSON para representar la información.
- SOAP utiliza XML como formato de datos, lo que puede ser más verboso y pesado en comparación con JSON.

### **4.3.1.3. Sencillez y Facilidad de Uso**

- REST es más simple de entender y utilizar debido a su enfoque en recursos y métodos HTTP estándar.
- SOAP puede ser más complejo debido a su estructura XML y la necesidad de definir esquemas XML específicos.

## **4.3.2. Principales diferencias entre REST y SOAP**

Ahora, profundicemos en algunas de las diferencias clave:

### **4.3.2.1. Protocolo vs Protocolo-Independiente**

- REST es un estilo de arquitectura basado en el protocolo HTTP, mientras que SOAP puede utilizar diferentes protocolos de transporte.

### **4.3.2.2. Formato de Datos**

- REST utiliza formatos de datos más ligeros como JSON, que son más fáciles de parsear y consumir.
- SOAP utiliza XML, que es más extensible pero puede ser más pesado y difícil de procesar.

### **4.3.2.3. Uso de Metodos HTTP**

- REST utiliza métodos HTTP estándar como GET, POST, PUT y DELETE para realizar operaciones en recursos.
- SOAP utiliza un conjunto definido de operaciones como “Request-Response” o “One-Way,” independientemente de los métodos HTTP.

### **4.3.3. Consideraciones de rendimiento y seguridad**

En cuanto al rendimiento, REST tiende a ser más eficiente debido a su uso de formatos de datos más ligeros y métodos HTTP estándar. Además, REST es más adecuado para aplicaciones web que requieren alta escalabilidad.

En cuanto a la seguridad, ambas tecnologías pueden ser seguras, pero SOAP a menudo se considera más seguro debido a su soporte integrado para la seguridad a través de WS-Security.

#### 4.3.4. Selección del protocolo adecuado para el proyecto

Dentro del contexto del proyecto del sistema de peaje, es esencial considerar varias características clave al seleccionar el protocolo de comunicación más apropiado. Una de las características más críticas a tener en cuenta es la escalabilidad, que desempeña un papel crucial en el éxito del sistema de peaje, dado su potencial para manejar un gran volumen de transacciones.

En este escenario, el protocolo REST se presenta como una opción sobresaliente. La razón principal es su capacidad inherente para escalar de manera efectiva.

Además de su capacidad de escalabilidad, REST también ofrece la ventaja adicional de permitir la construcción de API (Interfaces de Programación de Aplicaciones) de manera sencilla. Dado que el desarrollo de API es una parte integral del proyecto del sistema de peaje, el uso de REST facilita la creación y gestión de estas interfaces. Esto proporciona una mayor flexibilidad y modularidad al sistema, lo que puede simplificar el mantenimiento y la expansión futura.

### 4.4. Front-end

En el contexto del desarrollo web, el “Front-end” se refiere a la parte de una aplicación web que interactúa directamente con el usuario, es decir, la interfaz de usuario visible y las funcionalidades que se ejecutan en el navegador web del usuario.

#### 4.4.1. Uso de JavaScript, HTML, CSS o Bootstrap y Thymeleaf

- **JavaScript** es un lenguaje de programación ampliamente utilizado en el desarrollo web para agregar interactividad y dinamismo a las páginas web.
- **HTML** (HyperText Markup Language) es el lenguaje de marcado estándar utilizado para crear la estructura y el contenido de las páginas web.
- **CSS** (Cascading Style Sheets) es un lenguaje utilizado para dar estilo y diseño a las páginas web.
- **Bootstrap** es un popular framework de CSS y JavaScript que simplifica el desarrollo front-end<sup>[9]</sup>. Proporciona una serie de componentes y estilos predefinidos que facilitan la creación de interfaces responsivas y atractivas.
- **Thymeleaf** es un motor de plantillas que se utiliza para la generación de contenido dinámico en aplicaciones web. A diferencia de JavaScript, HTML, CSS o Bootstrap, que se centran en el lado del cliente, Thymeleaf opera en el lado del servidor. Esto significa que Thymeleaf permite la ejecución de código del lado del servidor para generar dinámicamente partes de una página web antes de enviarla al cliente.

#### 4.4.2. React vs Angular

React y Angular son dos populares frameworks/librerías de JavaScript utilizados para crear aplicaciones web <sup>[10]</sup> <sup>[11]</sup>. Se proporcionan las ventajas y desventajas de cada uno:

#### 4. HERRAMIENTAS Y TECNOLOGÍAS

---

##### 4.4.2.1. React

Analisis de ventajas y desventajas de React:

Ventajas	Desventajas
React es una librería de JavaScript que se centra en la creación de interfaces de usuario (UI) y se integra fácilmente con otras bibliotecas o frameworks	Puede llevar tiempo familiarizarse con los conceptos y herramientas asociados con React.
React utiliza un Virtual DOM para optimizar las actualizaciones de la interfaz de usuario, lo que mejora el rendimiento en aplicaciones de gran escala	React deja muchas decisiones de arquitectura a los desarrolladores, lo que puede ser desafiante para equipos sin experiencia
React tiene una comunidad grande y activa, lo que significa que hay muchas bibliotecas y recursos disponibles	
React permite a los desarrolladores elegir otras bibliotecas y herramientas para tareas adicionales como manejo de estado (Redux) y enrutamiento (React Router)	

**Tabla 4.2:** Ventajas y desventajas React

##### 4.4.2.2. Angular

Analisis de ventajas y desventajas de Angular:

Ventajas	Desventajas
Angular es un framework completo que proporciona una estructura y herramientas más definidas para el desarrollo de aplicaciones web	La complejidad de Angular puede resultar abrumadora para proyectos pequeños o equipos con menos experiencia
Angular ofrece una sólida inyección de dependencias y un sistema de módulos que facilita la organización del código	Las aplicaciones Angular tienden a requerir más código en comparación con React para realizar tareas similares
Angular utiliza TypeScript, que brinda ventajas como tipado estático y autocompletado en el desarrollo	Las actualizaciones de Angular son menos frecuentes que las de React, lo que puede llevar a utilizar versiones antiguas durante más tiempo
Angular CLI simplifica la creación y gestión de proyectos Angular	

**Tabla 4.3:** Ventajas y desventajas Angular

##### 4.4.3. Ejemplos de casos de uso para React y Angular

Se proporcionan ejemplos de casos de uso para React y Angular:

#### 4.4.3.1. React

- **Aplicaciones de una sola página (SPA):** React es una elección sólida para SPAs donde se necesita una actualización dinámica de la interfaz de usuario sin recargar la página.
- **Aplicaciones móviles:** Con React Native, se puede expandir para desarrollar aplicaciones móviles nativas.

#### 4.4.3.2. Angular

- **Aplicaciones empresariales:** Angular se adapta bien a aplicaciones empresariales que requieren una estructura sólida y una gestión clara de la inyección de dependencias.
- **Aplicaciones grandes y complejas:** Para proyectos que involucran equipos grandes y complejos, Angular ofrece una estructura organizativa y un sistema de módulos que facilitan la colaboración.
- **Aplicaciones que requieren tipado estático:** Si la seguridad y el control sobre el código son fundamentales, Angular con TypeScript proporciona un ambiente seguro.

#### 4.4.4. Selección del Framework Adecuado para el Proyecto

En el contexto del proyecto del sistema de peaje, hemos decidido optar por React como el framework en el frontend. Esta elección se basa en una serie de consideraciones clave que hacen que React sea la opción más adecuada.

React se destaca por su gran flexibilidad, lo que lo hace especialmente apropiado para un sistema de peaje que requiere escalabilidad. Su enfoque en la construcción de componentes modulares y la actualización eficiente del DOM (Modelo de Objeto del Documento) se alinea bien con los requisitos de una aplicación en constante crecimiento como un sistema de peaje.

React se ha ganado una reputación de ser intuitivo y fácil de aprender. Además, es uno de los frameworks más utilizados en la actualidad, respaldado por una gran comunidad y una amplia variedad de recursos disponibles.

### 4.5. Back-End

El “Back-End” se refiere a la parte de una aplicación web que se encarga de la lógica y el procesamiento de los datos o logica de negocio. En otras palabras, es la parte de la aplicación que se ejecuta en el servidor y se comunica con la base de datos.

#### 4.5.1. Uso de Java en el proyecto

Java es un lenguaje de programación ampliamente reconocido por su robustez, seguridad y portabilidad, lo que lo convierte en una elección ideal para el desarrollo en el back-end.

## 4. HERRAMIENTAS Y TECNOLOGÍAS

---

### 4.5.2. Uso de Spring Framework en el proyecto

Spring ofrece una arquitectura sólida y modular que simplifica significativamente el desarrollo de aplicaciones web [12]. Además, cuenta con una amplia gama de proyectos relacionados con Spring que amplían el framework para abordar necesidades específicas de funcionalidades.

### 4.5.3. Distintos proyectos Spring

Spring Framework consta de varios proyectos y módulos que abordan necesidades específicas en el desarrollo de aplicaciones. Algunos de los proyectos más conocidos son:

- **Spring Data:** Proporciona un enfoque coherente y simplificado para acceder y gestionar datos en diversas fuentes de datos, incluyendo bases de datos relacionales y NoSQL.
- **Spring Security:** Se centra en la autenticación y la autorización.
- **Spring Web:** Incluye módulos como Spring MVC para el desarrollo de aplicaciones web y Spring WebFlux para aplicaciones reactivas.
- **Spring Boot:** Facilita la creación rápida de aplicaciones Spring con configuración automática y un conjunto de bibliotecas preconfiguradas.

### 4.5.4. Configuración y módulos relevantes de Spring

La configuración en Spring se realiza a través de anotaciones, archivos XML o propiedades externas.

## 4.6. MySQL Base de Datos

MySQL es un sistema de gestión de bases de datos relacionales ampliamente reconocido que simplifica el almacenamiento y la administración eficiente de datos[13]. Como mencionamos previamente, hemos optado por SQL como el sistema de almacenamiento de datos, lo que se integra de manera ideal con el sistema MySQL. La elección de MySQL se fundamenta en su facilidad de uso y su capacidad para escalar en sistemas de gran envergadura.

## 4.7. Python

El uso de Python en el proyecto se contempla en el caso del incremento 5 del proyecto.

### 4.7.1. Áreas específicas donde se utilizará Python

Python se utilizará para el reconocimiento de matrículas en el módulo IPD. Python tiene bibliotecas y herramientas como OpenCV que son ampliamente utilizadas para tareas de visión por computadora, lo que facilita la implementación del reconocimiento de matrículas.

#### 4.7.2. Razones para considerar el uso de Python en el proyecto

Python es ampliamente reconocido por su capacidad de integración con otros lenguajes y tecnologías [14]. Esto nos permite expandir nuestras posibilidades, desde la creación de algoritmos personalizados para el reconocimiento de matrículas hasta el web scrapping de etiquetas ambientales.

### 4.8. Snapshot API de Plate Recognizer

#### 4.8.1. Descripción de la Snapshot API de Plate Recognizer

Snapshot API de Plate Recognizer<sup>1</sup> es una API diseñada para realizar el reconocimiento automático de matrículas vehiculares a partir de imágenes o instantáneas (snapshots). Esta API se basa en tecnologías de visión por computadora y aprendizaje profundo para analizar las imágenes y extraer información de las matrículas de los vehículos presentes en ellas.

#### 4.8.2. Uso de la API para reconocimiento de matrículas

La utilización de la API de Plate Recognizer en el proyecto permite que la aplicación realice tareas como:

- Identificar y registrar matrículas de vehículos en un sistema de peaje.
- Realizar análisis de tráfico y conteo de vehículos.

#### 4.8.3. Beneficios y aplicaciones de Plate Recognizer en el proyecto

Los beneficios de utilizar Plate Recognizer en el proyecto incluyen:

- Mayor eficiencia en la identificación de matrículas, lo que mejora los procesos que dependen de esta información.
- Reducción de errores humanos en la identificación de matrículas, especialmente relevante en sistemas de peaje y control de acceso.
- Automatización de tareas que normalmente requerirían intervención manual, lo que resulta en ahorro de tiempo y recursos.
- Fortalecimiento de la gestión de seguridad y acceso al identificar vehículos autorizados y no autorizados.
- Recopilación de datos valiosos sobre el flujo de vehículos, que pueden utilizarse para análisis y toma de decisiones.

#### 4.8.4. Desventajas de Plate Recognizer en el proyecto

- Dependiendo de la cantidad de solicitudes o imágenes procesadas, el uso de la API puede generar costos adicionales. Es importante evaluar el costo-beneficio en relación con el volumen de uso.

---

<sup>1</sup><https://platerecognizer.com/>

#### **4. HERRAMIENTAS Y TECNOLOGÍAS**

---

- La API requiere una conexión a Internet para funcionar. Si la conectividad falla o es inestable, puede afectar la capacidad de la aplicación para realizar el reconocimiento de matrículas.
- La integración de una API externa implica una dependencia de terceros. Si la API experimenta interrupciones o cambios en sus políticas, podría afectar la funcionalidad del proyecto.
- El reconocimiento de matrículas a través de una API externa puede introducir cierta latencia en la respuesta de la aplicación, especialmente si hay un alto volumen de solicitudes o si la API experimenta congestión.
- La integración y el mantenimiento de una API externa requieren tiempo y esfuerzo. Es importante estar preparado para gestionar actualizaciones y cambios en la API a lo largo del tiempo.

### **4.9. Testeo de Código: Postman, JUnit y Mockito**

#### **4.9.1. JUnit y AssertJ**

JUnit es un marco de trabajo (framework) de pruebas unitarias para Java. Se utiliza para escribir y ejecutar pruebas automatizadas en el código Java del proyecto. Las pruebas unitarias se centran en verificar que las unidades individuales de código (como métodos o funciones) funcionen correctamente. JUnit proporciona anotaciones y clases para definir y ejecutar estas pruebas de manera eficiente.

AssertJ es una biblioteca de aserciones (assertions) para Java que proporciona una sintaxis más legible y expresiva para realizar afirmaciones o comprobaciones en las pruebas unitarias de código Java. Su objetivo principal es mejorar la claridad y la facilidad de mantenimiento de las pruebas al proporcionar una amplia gama de métodos de aserción que son intuitivos y autodescriptivos.

#### **4.9.2. Mockito**

Mockito es una biblioteca de Java que facilita la creación y el uso de objetos simulados (mocks) en las pruebas unitarias. Estos mocks se utilizan para simular el comportamiento de objetos reales y controlar el entorno de prueba. Mockito permite crear mocks de manera sencilla y definir cómo deben comportarse durante las pruebas, lo que facilita la simulación de dependencias y la realización de pruebas unitarias más efectivas.

#### **4.9.3. Uso de Postman en el proyecto**

Postman es una herramienta de desarrollo y pruebas de API que se utiliza para crear, ejecutar y automatizar pruebas de API. En el proyecto, Postman se emplea para realizar pruebas de integración y asegurar que las API funcionen correctamente. Esto implica la creación y ejecución de solicitudes HTTP/API a los servicios web para verificar su funcionamiento y obtener respuestas adecuadas.

#### 4.9.3.1. Creación y ejecución de solicitudes HTTP/API

Con Postman, se pueden crear solicitudes HTTP como GET, POST, PUT o DELETE y enviarlas a las API para probar su funcionalidad. Se pueden proporcionar datos de entrada, encabezados y parámetros de consulta, y Postman registrará las respuestas recibidas. Esto permite validar que las rutas y funcionalidades de las API estén configuradas y respondan adecuadamente.

#### 4.9.3.2. Automatización de pruebas y documentación de API

Una ventaja adicional de Postman es su capacidad para automatizar pruebas. Se pueden crear colecciones de solicitudes y definir flujos de trabajo de prueba que se ejecutan automáticamente. Esto es especialmente útil para pruebas de regresión, donde se verifica que las modificaciones de código no afecten las funcionalidades existentes.

Además de las pruebas, Postman también permite documentar las API de manera efectiva. Se puede crear documentación detallada para los servicios web, incluyendo ejemplos de solicitudes y respuestas, lo que facilita la comprensión y el uso de las API por parte de otros desarrolladores.

### 4.10. Logging: Log4j2

Log4j2 es una biblioteca de registro (logging) para aplicaciones Java que se utiliza para capturar y gestionar registros (logs) durante la ejecución de una aplicación. Permite registrar eventos, mensajes y errores en diferentes niveles de severidad. Log4j2 ofrece configuración flexible y diversas opciones de salida de registros, como archivos de registro, consola y servicios de registro remoto. Esto facilita el diagnóstico de problemas, la monitorización y la depuración de aplicaciones, lo que es esencial para mantener la calidad y la confiabilidad del software.

### 4.11. Project Lombok

Project Lombok es una herramienta que simplifica y acelera el desarrollo de aplicaciones en Java al eliminar la necesidad de escribir código repetitivo y boiler-plate [15]. Al hacerlo, permite a los desarrolladores centrarse en la creación de componentes críticos y no perder tiempo en partes innecesarias y redundantes del código.

### 4.12. DevOps

DevOps es una metodología que combina las disciplinas de desarrollo de software y operaciones de TI para lograr una entrega de software más rápida, eficiente y confiable. Se basa en la automatización de procesos, la colaboración estrecha entre equipos y la búsqueda constante de mejoras en todas las etapas del ciclo de vida del software.

#### 4.12.1. Jenkins

Jenkins es una herramienta de código abierto que se utiliza para automatizar la construcción, el despliegue y la gestión de proyectos de software. Se integra con varias herramientas

#### **4. HERRAMIENTAS Y TECNOLOGÍAS**

---

y servicios para orquestar pipelines de entrega continua. En el contexto de DevOps, Jenkins desempeña un papel fundamental al permitir la automatización de procesos, la integración continua y la entrega continua, lo que facilita la colaboración y la entrega eficiente de software.

##### **4.12.2. Docker**

Docker es una plataforma que permite a los desarrolladores empacar, distribuir y ejecutar aplicaciones y sus dependencias en contenedores ligeros y portátiles [16]. Los contenedores son entornos aislados que incluyen todo lo necesario para que una aplicación funcione, como bibliotecas, dependencias y configuraciones. Docker simplifica el despliegue y la gestión de aplicaciones, ya que garantiza que las aplicaciones se ejecuten de la misma manera en cualquier entorno, desde el desarrollo hasta la producción.

Docker se puede utilizar de manera opcional para lograr una arquitectura de microservicios. Esto implica dividir la aplicación en componentes más pequeños y autónomos, cada uno ejecutándose en su propio contenedor Docker.

# 5

## CAPÍTULO

# Arquitectura del Sistema

Después de haber descrito las tecnologías a utilizar en el capítulo anterior. En este capítulo, se presentará el sistema en general, incluyendo el conjunto de tecnologías utilizado y la arquitectura del sistema.

## 5.1. Conjunto de tecnologías utilizadas

En esta sección se presenta el conjunto de tecnologías utilizado en las diferentes capas:

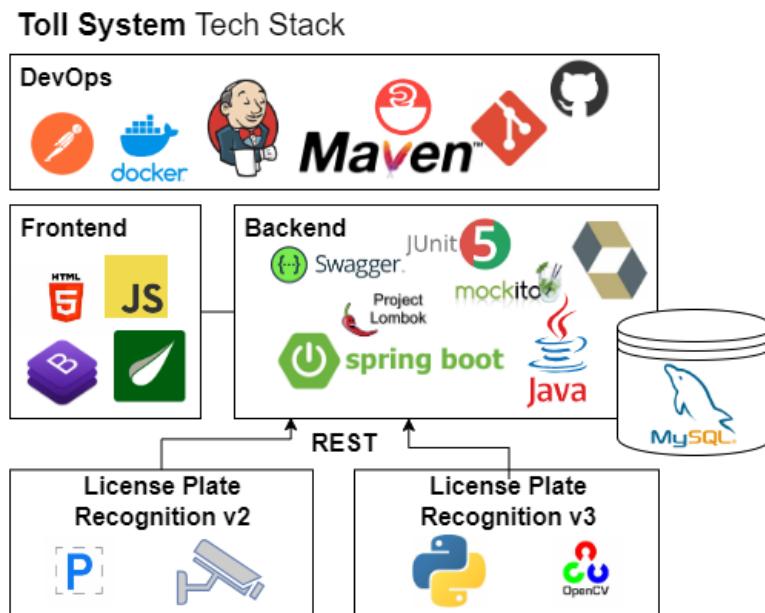


Figura 5.1: Conjunto de tecnologías utilizados

En el sistema existen 5 módulos de capas:

- **Capa de Presentación:** Thymeleaf para plantillas, JavaScript para lógica del cliente, HTML para estructura, CSS para estilo y Bootstrap para diseño.

## 5. ARQUITECTURA DEL SISTEMA

- **Capa de Aplicación:** Swagger para documentación de API, Spring Boot para desarrollo rápido, Project Lombok para código limpio, Java como lenguaje principal, Mockito y JUnit para pruebas.
- **Capa de Reconocimiento de Matrícula:** Tres versiones de API de reconocimiento de matrículas, incluyendo Plate Recognizer y un algoritmo propio.
  - La versión 1, que utilizaba la API de "Plate Recognizer", está obsoleta debido a un manejo incorrecto de los datos, específicamente en la integración de la imagen de procesamiento.
  - La versión 2 hace uso de la API de terceros "Plate Recognizer", mejorando el manejo de los datos de la versión 1.
  - Mientras que la versión 3 representa una expansión que incorpora nuestro propio algoritmo utilizando Python.
- **Capa de Persistencia:** Hibernate para mapeo objeto-relacional, MySQL como base de datos.
- **Capa de DevOps:** Log4j2 para registro de eventos, Postman para pruebas de API, Maven y Jenkins para automatización de tareas, Git y GitHub para control de versiones.

### 5.2. Modelo de flujo de datos del sistema

Para mejorar la comprensión de la relación entre las diversas tecnologías empleadas y cómo interactúan en el contexto del proyecto del sistema de peaje, hemos optado por elaborar una sección que detalla el modelo de flujo de datos del sistema. Aquí, se presenta una visión más clara de los distintos componentes del sistema y los servicios específicos que ofrecen. Esta descripción del flujo de datos es esencial para una comprensión más profunda de cómo se entrelazan las tecnologías clave en este proyecto de peaje.

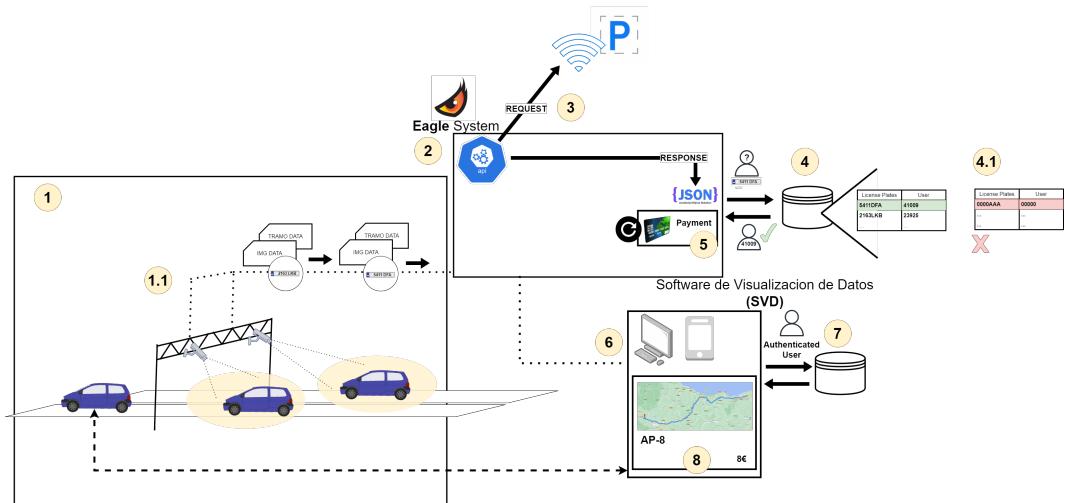


Figura 5.2: Modelo de flujo de datos del sistema

1. A lo largo del documento, se ha destacado que el sistema actual es simplificado. Sin embargo, mediante la representación visual en la imagen 5.2, buscamos ilustrar la realidad de la simplificación del sistema. La captura de imágenes a través de cámaras y la localización de los tramos se llevan a cabo mediante el sistema de administración definido previamente en el capítulo 3 de captura de requisitos. Esto enfatiza que esta parte no puede ejecutarse con los recursos actuales, lo que justifica la simplificación del proyecto y la realización del sistema de administración.
  - a) La información capturada, en forma de imágenes, se conserva, ya que desempeña un papel fundamental en el proceso de reconocimiento de matrículas. Lo que se transfiere en esta etapa es tanto la información de la imagen como los datos relativos al tramo actual, que se envían al sistema Eagle de procesamiento de datos.
2. El sistema Eagle, también conocido como IPD (Interfaz de Procesamiento de Datos), se encarga de llevar a cabo todo el procesamiento de datos relacionados con el reconocimiento de matrículas y los procesos de pago. Además, en este sistema, se gestiona la comunicación con la API de Plate Recognizer, que se utiliza en la actualidad para este propósito. Otra opción, es utilizar nuestra API que mantiene el nombre del modulo, Eagle.
3. La comunicación entre Plate Recognizer y nuestra API requieren una conexión a internet. En este proceso, se envía una solicitud con la imagen capturada y se aguarda una respuesta de la API en formato JSON. Esta respuesta se analiza para extraer la información necesaria y así poder obtener los datos del usuario que está conduciendo el vehículo.
4. La información del usuario se obtiene a partir de la respuesta proporcionada por las APIs mencionadas anteriormente.
5. El proceso de pago se realiza en función del tipo de vehículo y etiqueta ambiental y la información relativa al tramo en cuestión.
6. El Software de Visualización de Datos (SVD) tiene la responsabilidad de presentar los datos a los usuarios finales del sistema, quienes pueden acceder a ellos desde sus dispositivos móviles u ordenadores.
7. Los usuarios autenticados e identificados tienen la capacidad de acceder a la información relacionada con su cuenta desde sus dispositivos, lo que les permite gestionar aspectos como pagos.
8. Adicionalmente, se presentan los datos del tramo, incluyendo información sobre el viaje, como la carretera utilizada, el origen y destino, el costo asociado, y otros detalles relevantes para una experiencia completa.

### 5.3. Capas del sistema

En esta sección, se explican las capas del sistema y la arquitectura del proyecto

El sistema consta de 4 capas:

## 5. ARQUITECTURA DEL SISTEMA

---

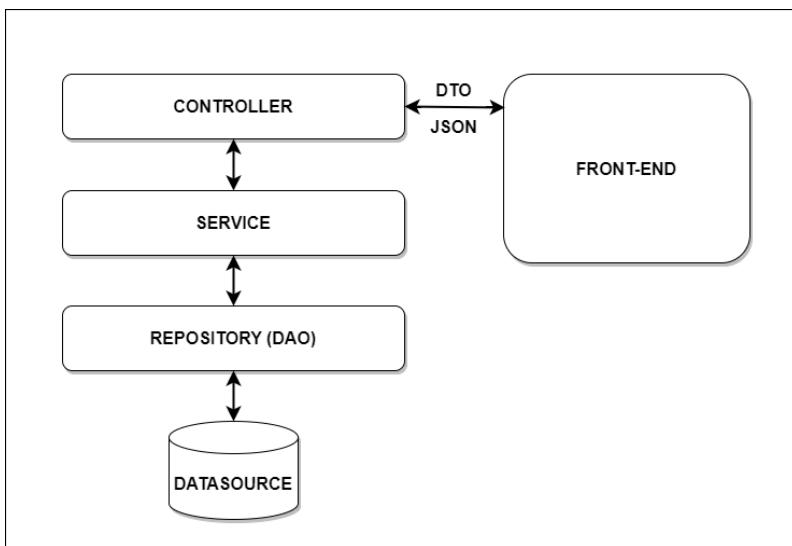


Figura 5.3: Capas del sistema

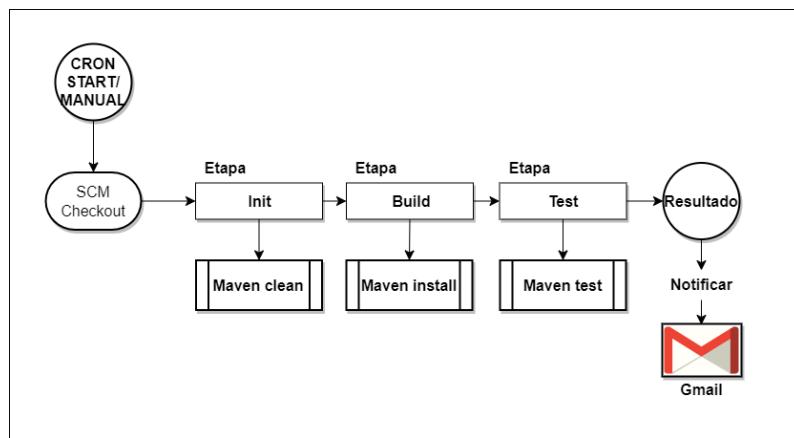
1. **Capa del Controlador:** En esta capa se establece la conexión con el frontend del sistema y también se denomina capa de presentación o API, ya que transfiere los objetos DTO (Data Transfer Objects) o JSON al frontend y otros tipos de solicitudes con el frontend del sistema. Esta capa facilita la representación de las consultas HTTP entre el cliente y el servidor para lograr una comunicación efectiva.
2. **Capa de Servicio:** En esta capa, se encuentra la lógica de negocio principal de la aplicación. Los servicios son responsables de ejecutar operaciones específicas, coordinar la interacción entre diferentes componentes y garantizar que la lógica de negocio se implemente de manera coherente. Los servicios suelen ser invocados desde la capa de controlador y actúan como intermediarios entre la interfaz de usuario y la capa de datos.
3. **Capa de Repositorio o DAO (Data Access Object):** Esta capa se encarga de interactuar con la capa de datos para realizar operaciones de lectura y escritura en la base de datos. Utiliza patrones de acceso a datos como DAO para abstraer la lógica de acceso a datos, lo que facilita la manipulación de la información almacenada. Los repositorios actúan como intermediarios entre la capa de servicio y la capa de datos, permitiendo que la lógica de negocio se concentre en los servicios.
4. **Capa de Datasource o Base de datos:** Por último, esta capa se encuentra la propia base de datos o el origen de los datos. Aquí es donde se almacenan y recuperan los datos que la aplicación utiliza. Puede ser una base de datos relacional, no relacional u otro tipo de almacenamiento de datos. La capa de datos se encarga de gestionar la persistencia de la información y proporcionar interfaces para que las capas superiores puedan acceder y manipular los datos de manera eficiente.

En el contexto del sistema, el Front-end hace referencia al módulo SVD, que a su vez presenta sus propias capas. En este módulo, los usuarios pueden interactuar y visualizar los datos del sistema. Contrariamente, el módulo IPD prescinde de Front-end. Esta elección

se basa en la decisión previa de simularlo de manera diferente, debido a su naturaleza considerablemente más compleja en comparación con el módulo SVD.

## 5.4. CI/CD pipeline con Maven y Docker

En esta sección, se abordará en detalle el pipeline diseñado para el flujo de CI/CD utilizando Jenkins. Para obtener una comprensión completa del proceso, se recomienda consultar el Capítulo 5.8, titulado "Lecciones aprendidas". En este capítulo, se proporciona una explicación detallada sobre la instalación de las herramientas necesarias y se abordan los desafíos y problemas encontrados durante el desarrollo.



**Figura 5.4:** CI/CD pipeline

El pipeline se compone de tres etapas: Init, Build y Test, cada una con su correspondiente conjunto de tareas. Como se menciona, el pipeline creado tiene un diseño sencillo con el propósito de explorar las posibilidades para establecer un flujo de integración continua.

Estas etapas están especificadas en un archivo especial llamado `Jenkinsfile`. A continuación, se detalla cada una de las etapas representadas en la figura 5.4:

1. **SCM Checkout:** Esta etapa es interna del pipeline y tiene como objetivo obtener la última versión de la rama en el repositorio Git del proyecto.
2. **Init:** En esta fase de inicialización del pipeline, se realiza la limpieza de la construcción anterior del código.
3. **Build:** En esta etapa, se procede a construir el código sin ejecutar las pruebas.
4. **Test:** Aquí se lleva a cabo la fase de prueba del código.
5. **Notif:** Aunque esta etapa no se muestra explícitamente, se ejecuta después de completar las etapas para notificar si el pipeline se ejecutó con éxito o si ha fallado.

## 5.5. Configuración de Producción y Desarrollo

En esta sección, detallamos la configuración realizada para los entornos de producción y desarrollo. Dado que empleamos Spring, en el archivo `application.properties` podemos definir los perfiles de ejecución.

En nuestro contexto, hemos establecido dos perfiles: `prod` y `dev`, cada uno con su respectivo archivo:

- `application-prod.properties`
- `application-dev.properties`

Spring busca automáticamente el archivo correspondiente y sobrescribe la configuración común con las definiciones específicas. De esta manera, tenemos un control preciso sobre las ejecuciones de los entornos. En nuestro caso, esta práctica se limita al control de acceso a la base de datos, ya que contamos con dos bases de datos: una para pruebas y desarrollo en `dev`, y otra de producción en `prod`. Este enfoque nos permite gestionar de manera estricta la ejecución de pruebas, servicios y otras configuraciones necesarias.

## 5.6. Seguridad en el sistema

En esta sección, se proporciona una explicación detallada sobre la seguridad en el sistema, abordando tanto la seguridad en los servicios REST como la seguridad en la aplicación y el simulador.

Además, se describe el proceso de anonimización de los datos obtenidos a través de los servicios REST, los cuales se encuentran protegidos mediante el método K-Anonimización, empleando la técnica de enmascaramiento de datos de los usuarios para garantizar el más alto nivel de seguridad.

### 5.6.1. Seguridad de Usuario y Simulador

La aplicación y el simulador son dos componentes distintos y esenciales del sistema, cada uno desempeñando un papel único en su funcionamiento. El simulador tiene como propósito principal la simulación del comportamiento de los datos del sistema de peaje, mientras que la aplicación permite a los usuarios interactuar con los datos, visualizarlos, realizar pagos y acceder a los servicios REST. Ambos módulos, el SVD y el IPD, son fundamentales para el funcionamiento completo del sistema.

La seguridad desempeña un papel crítico en este contexto, ya que sin medidas de seguridad adecuadas, la aplicación no puede funcionar de manera efectiva ni garantizar la protección de los datos sensibles. En este sentido, se implementa una estrategia de seguridad sólida. Se utiliza una técnica de seguridad "simple" que aprovecha la mejor función hash conocida y ampliamente utilizada hasta el momento: BCrypt.

BCrypt es una función hash unidireccional, lo que significa que no es posible revertir el proceso de hash y recuperar la contraseña original a partir de su hash. Esta característica es esencial para proteger las contraseñas de los usuarios de manera efectiva y garantizar la confidencialidad de sus datos.

### 5.6.2. Seguridad de los servicios REST

La seguridad de los servicios REST es de vital importancia, ya que no se puede permitir un acceso no controlado a los mismos. Aunque en el contexto de la aplicación no se manejan datos personales de los usuarios, es esencial explorar la utilización y protección de estos servicios mediante el uso de tokens JWT (JSON Web Tokens).

Este método implica la generación de un token único para cada usuario, lo que les permite acceder y utilizar los servicios REST de manera autenticada. Sin un token válido, los servicios REST no pueden ser utilizados, lo que garantiza una consistencia en la seguridad y evita accesos no autorizados. Aunque no se trate de datos personales en el contexto de la aplicación, esta medida fortalece la seguridad y controla el acceso a los servicios, contribuyendo así a la protección de los recursos del sistema.

### 5.6.3. K-Anonimización

La técnica de K-Anonimización es ampliamente reconocida en el ámbito de la anonimización de datos, y su aplicación es crucial en nuestro contexto del sistema. Durante el uso de nuestros servicios REST, se recopila información de los usuarios, lo que plantea la necesidad de abordar la protección de estos datos de manera efectiva.

Para resolver este desafío, hemos implementado una técnica de enmascaramiento de datos que oculta información sensible, específicamente tres de los siete caracteres de las matrículas de vehículos. Por ejemplo, si se presenta la matrícula "1234ABC", la K-Anonimización la transformará de la siguiente manera: "\*2\*\*ABC".

Este enfoque permite ocultar información identificable de los usuarios y garantiza la privacidad de los datos. Esta técnica se emplea con el propósito de visualizar la información de manera anónima, contribuyendo así a la protección de la privacidad de los usuarios.

## 5.7. Cambios en la estructura del proyecto

En esta sección se aclara la estructura del proyecto, lo que permite detallar la estructura final del mismo y aclarar que se había planificado inicialmente con el objetivo de mejorar la visión general. Se utiliza la estandarización de proyectos Maven para que cualquier persona pueda gestionar y comprender la ubicación del código de manera efectiva.

El proyecto se divide en dos paquetes iniciales: eus.ehu.tfg.ktoda.tollinit y eus.ehu.tfg.ktoda.tollsystem. Aquí está la descripción de cada uno:

- **eus.ehu.tfg.ktoda.tollinit:** Este paquete se ha creado con el propósito de inicializar datos y realizar pruebas en el entorno de desarrollo.
- **eus.ehu.tfg.ktoda.tollsystem:** En este paquete se encuentran todos los elementos fundamentales de la aplicación, incluyendo los dos módulos, las entidades, la seguridad y la configuración necesaria.
  - **model:** Contiene las clases que definen la estructura de datos de la aplicación.
  - **eagle.api:** Aquí se encuentra la interfaz y la implementación de la API relacionada con el modulo IPD o Eagle.

## 5. ARQUITECTURA DEL SISTEMA

---



Figura 5.5: Estructura del proyecto actual

- config: Incluye la configuración de la aplicación, como archivos de configuración y clases de configuración.
- controller: Contiene controladores que manejan las solicitudes HTTP y gestionan las respuestas.
- repository: Aquí se definen y gestionan las consultas a la base de datos.
- service: Contiene la lógica de negocio de la aplicación.
- security: Se ocupa de aspectos relacionados con la seguridad de la aplicación.
- util: Contiene utilidades y funciones de propósito general utilizadas en la aplicación.

Además, se hace uso de un conjunto de datos de imágenes llamado toll-imgs, el cual se emplea en las pruebas de imágenes, tal como se definió previamente.

Después de la quinta reunión, como se detalla en la referencia 12.4, se tomó la decisión de modificar la estructura del proyecto. Esta decisión se basó en la comprensión más clara del proyecto, lo que condujo a la consolidación de la estructura definitiva del mismo.

La estructura elegida se basa en características específicas, lo que proporciona varios beneficios:

## 5.7. Cambios en la estructura del proyecto



Figura 5.6: Estructura del proyecto final

- **Alta Cohesión:** Los componentes relacionados se agrupan de manera lógica, lo que facilita el mantenimiento y comprensión del código.
- **Bajo Acoplamiento:** Los componentes están débilmente acoplados, lo que significa que los cambios en un componente tienen un impacto mínimo en otros, mejorando la robustez y la capacidad de cambio.
- **Alta Modularidad:** El sistema se divide en módulos independientes, lo que facilita la reutilización de código y la gestión de complejidad.
- **Encapsulación de Clases Concretas:** Se restringe el acceso público a clases solo a aquellas que son necesarias, lo que promueve un diseño más seguro y controlado.
- **Facilita la Transición de Aplicación Monolítica a Microservicios:** La estructura es adaptable para migrar de una arquitectura monolítica a microservicios, lo que mejora la escalabilidad y la flexibilidad.
- **Apoya el Diseño de Dominio Orientado a Objetos:** Facilita la representación del dominio del problema en el código, lo que mejora la comprensión y el mantenimiento a largo plazo.

- **Facilita las Pruebas Unitarias:** La concentración en características específicas simplifica las pruebas de cada componente, lo que mejora la calidad del código y la detección temprana de errores.

## 5.8. Lecciones aprendidas

### 5.8.1. Creación de CI/CD Pipeline con Jenkins en Docker

1. **Contexto:** Al enfrentarse a proyectos de esta envergadura, es beneficioso explorar la implementación de un flujo de trabajo de integración continua que automatice tanto la construcción como las pruebas de código. En este caso, se optó por utilizar Jenkins de manera aislada en un entorno Docker para lograr la eficiencia deseada en dicho flujo de trabajo.
2. **Objetivos:** Los objetivos del pipeline incluyeron la automatización al realizar un push o pull request en la rama principal del repositorio Git del proyecto. Asimismo, se buscó establecer un pipeline sencillo que abarcara la inicialización, construcción, prueba y notificación de manera eficiente.
3. **Configuración de Jenkins en Docker:** A continuación, se detalla el proceso para la creación del pipeline y toda la configuración. Nota: toda la información está disponible en el repositorio del proyecto, específicamente en el archivo README.md de la carpeta ci-cd.
  - Antes de comenzar con la configuración, se debían cumplir varios requisitos:
    - Tener Docker instalado.
    - Tener Java instalado.
  - Luego, abrimos el terminal, utilizando WSL (Windows Subsystems for Linux) para aprovechar el entorno de trabajo Linux.
  - Creamos una red de puente Docker: `docker network create jenkins`
  - Ejecutamos la imagen Docker:

```
docker run --name jenkins-docker --rm --detach \
--privileged --network jenkins --network-alias docker \
--env DOCKER_TLS_CERTDIR=/certs \
--volume jenkins-docker-certs:/certs/client \
--volume jenkins-data:/var/jenkins_home \
--publish 2376:2376 \
docker:dind
```

- Personalizamos la imagen oficial de Docker de Jenkins: Creamos un archivo Dockerfile con el siguiente contenido:

```
FROM jenkins/jenkins:2.414.2-jdk17
USER root
RUN apt-get update && apt-get install -y lsb-release
RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc \
```

```

https://download.docker.com/linux/debian/gpg
RUN echo "deb [arch=$(dpkg --print-architecture) \
signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
https://download.docker.com/linux/debian \
$(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
RUN apt-get update && apt-get install -y docker-ce-cli
USER jenkins
RUN jenkins-plugin-cli --plugins "blueocean docker-workflow"

```

- Construimos una nueva imagen Docker desde el archivo Dockerfile y le asignamos un nombre significativo, por ejemplo, "jenkins-docker-v213": docker build -t <nombre de la imagen>.
- Ejecutamos nuestra propia imagen <nombre de la imagen> como contenedor en Docker:

```

docker run --name jenkins-blueocean --restart=on-failure --detach \
--network jenkins --env DOCKER_HOST=tcp://docker:2376 \
--env DOCKER_CERT_PATH=/certs/client --env DOCKER_TLS_VERIFY=1 \
--volume jenkins-data:/var/jenkins_home \
--volume jenkins-docker-certs:/certs/client:ro \
--publish 8081:8080 --publish 50000:50000 <nombre de la imagen>

```

- En nuestro caso, cambiamos 8080:8080 a 8081:8080 para alojarlo en 8081 en la máquina local porque la aplicación principal se ejecuta en el puerto 8080.
- Abrimos Jenkins en localhost en el puerto especificado. Verificamos si el contenedor está en ejecución con docker ps. Si no está en ejecución, lo iniciamos con docker start <nombre de la imagen>.
- La primera vez que abrimos Jenkins, solicitará la configuración. Para obtener la contraseña de administrador de Jenkins, ejecutamos docker logs <nombre de la imagen>. Creamos nuestro usuario en Jenkins y ya podemos comenzar a realizar flujos de integración continua.

#### 4. Definición del Pipeline:

Una vez que hemos obtenido acceso a Jenkins, procedemos con la creación del flujo de integración continua.

- La estructura del pipeline es bastante simple. Dado que no desplegamos el proyecto en un entorno de producción, se elimina una parte del flujo. Sin embargo, veremos que esta decisión conlleva dificultades más adelante.
- Creamos el pipeline desde el repositorio de código de GitHub, específicamente desde la rama main. En la rama main, creamos el archivo Jenkinsfile que contiene el script del pipeline. Como se especificó anteriormente, las etapas del pipeline son: inicialización, construcción y pruebas; al finalizar, se notifica por correo electrónico el estado del trabajo.
- La automatización del pipeline se llevó a cabo de manera diferente, ya que no se pudo lograr el objetivo deseado con push y pull en la rama main. Por lo tanto, la automatización se realizó creando un trabajo CRON que ejecuta el flujo cada 15

## 5. ARQUITECTURA DEL SISTEMA

---

minutos. En un entorno de despliegue, esto no se realizaría, ya que la ejecución del CI/CD es costosa.

- Para lograr todo esto, fue necesario crear una clave SSH de despliegue del repositorio para conectarse con Jenkins. Esto se debió a que la API de autenticación con contraseña y nombre de usuario de GitHub está obsoleta.

**5. Problemas:** Como se mencionó anteriormente, no se pudo obtener el flujo del pipeline deseado principalmente. En esta sección, se enumeran los problemas que se han enfrentado y las dificultades:

- Primero, la automatización de CI/CD al realizar push o pull no se pudo llevar a cabo, ya que GitHub no puede acceder a una conexión a internet con localhost. Se intentó realizar con Ngrok para abrir una conexión http, pero tampoco funcionó. No se podía extender demasiado el tiempo de esta tarea, ya que no estaba planificada y ya conllevaba una parte significativa del desarrollo.
- Nos enfrentamos a problemas con la conexión de Jenkins con GitHub, principalmente debido a que no estaba especificado de manera clara cómo realizar la conexión y la creación de un pipeline con GitHub.

**6. Lecciones Aprendidas:**

- Se aprendió cómo aislar y crear Jenkins en Docker, así como la creación de un pipeline simple sin despliegue.
- Aunque no se logró el objetivo deseado principalmente, se logró la realización de un flujo simple y se preparó para un futuro despliegue.

# 6

## CAPÍTULO

# Interfaz de Procesamiento de Datos - IPD (Eagle)

En este capítulo, se aborda y se investiga la Interfaz de Procesamiento de Datos (IPD) o Eagle. Se ha denominado "Eagle" o "Águila" debido a su conexión con el sistema de reconocimiento de matrículas. La elección de este nombre se debe a la asombrosa visión que poseen las águilas, y esta relación se refleja en el sistema gracias a su capacidad para capturar imágenes de manera excepcional.

## 6.1. Preparación de Datos

En la reunión 12.4, durante la validación del capítulo de captura de requisitos, se discutió que cierta parte de los datos relacionados con el administrador podría suponerse, ya que su inclusión no contribuía significativamente al objetivo del proyecto. Como resultado, se tomó la decisión de hacer supuestos sobre los datos de creación, específicamente, los datos relacionados con el inicio del sistema:

- Section
- Point
- Fee
- GeoLocation

Estos datos se considerarán como si ya existieran o estuvieran almacenados previamente en la aplicación, ya que realmente representan objetos reales y no se tratarán como un caso específico de administración. Esto implica el supuesto de que los datos relacionados con los tramos existen o están disponibles antes de la implementación de la aplicación.

En efecto, estos datos han sido recopilados del sitio oficial de VIA-T, la entidad encargada de la infraestructura de telepeaje en autopistas<sup>1</sup>. De esta manera, con esta información,

---

<sup>1</sup><https://www.viat.es/>

## 6. INTERFAZ DE PROCESAMIENTO DE DATOS - IPD (EAGLE)

---

podemos preparar los datos necesarios para el sistema de la IPD. Es importante destacar que se deben manejar con precaución los siguientes datos, ya que no han sido recogidos de manera precisa, sino que representan una aproximación, particularmente en lo que respecta a los puntos de control de las secciones.

### 6.1.1. Puntos de control

A continuación, se proporcionará una descripción más detallada del proceso de preparación de los datos y de los datos generados específicamente para el sistema.

Sección	Puntos	Kilometro	Lat	Long
AP-8	Hendaye (Francia)	0	43.3681° N	-1.7859° W
	San Sebastián (Donostia)	20	43.3126° N	-1.9756° W
	Irun	3	43.3392° N	-1.7922° W
	Hondarribia	4	43.3684° N	-1.7897° W
	Renteria (Erreenteria)	12	43.3085° N	-1.9095° W
	Lezo	12	43.3157° N	-1.8787° W
	Pasaia (Pasajes)	15	43.3326° N	-1.9397° W
	Lasarte-Oria	25	43.2799° N	-2.0196° W
	Usurbil	25	43.2828° N	-2.0655° W
	Orio	34	43.2826° N	-2.1044° W
	Deba	55	43.2986° N	-2.3499° W
	Durango	85	43.1687° N	-2.6369° W
	Amorebieta-Etxano	90	43.2235° N	-2.7320° W
	Bilbao	120	43.2630° N	-2.9346° W
AP-68	Zaragoza	300	41.6488° N	-0.8891° W
	Logroño	136	42.4664° N	-2.4499° W
	Haro	90	42.5804° N	-2.8472° W
	Miranda de Ebro	70	42.6864° N	-2.9462° W
	Vitoria-Gasteiz	40	42.8467° N	-2.6716° W
AP-1	Burgos	145	42.3400° N	-3.6992° W
	Briviesca	110	42.5525° N	-3.3233° W
	Miranda de Ebro	75	42.6864° N	-2.9462° W
	Vitoria-Gasteiz	45	42.8467° N	-2.6716° W
	Eibar	0	43.1850° N	-2.4732° W

Tabla 6.1: Asociación entre Puntos y Secciones

---

En la tabla 6.1, se presentan las tres secciones y sus respectivos puntos asociados, junto con información geográfica esencial. Esta información constituye la base fundamental del sistema de secciones, ya que nos permite definir las ubicaciones exactas de las mismas en la autopista. Es importante destacar que estos datos representan una aproximación a la realidad, dado que estamos simulando el funcionamiento del sistema de peaje.

### 6.1.2. Tasas

A continuación, se presentan las secciones junto con las tasas ficticias por kilómetro recorrido.

Sección	Tasa (/km)	Categoría Vehículo
AP-8	0.15 €	Categoría 1
	0.25 €	Categoría 2
	0.35 €	Categoría 3
AP-68	0.10 €	Categoría 1
	0.17 €	Categoría 2
	0.25 €	Categoría 3
AP-1	0.18 €	Categoría 1
	0.30 €	Categoría 2
	0.40 €	Categoría 3

**Tabla 6.2:** Asociación entre Tasas y Secciones

Se presentan las tasas ficticias que se han establecido para los diferentes tramos de la autopista. Estas tasas varían según la categoría de vehículo, que se divide en tres categorías principales:

1. **Categoría 1:** Vehículos ligeros.
2. **Categoría 2:** Vehículos medianos.
3. **Categoría 3:** Vehículos pesados.

Además, estas tasas se aplican en función de la distancia recorrida en kilómetros. La fórmula utilizada para calcular las transacciones es la siguiente:

$$|km\_entrada - km\_salida| * tasa\_seccion\_categoria * sticker_{co2}$$

La ecuación se expresa de la siguiente manera: se resta la posición de entrada en la autopista de la posición de salida en la sección para obtener la distancia en kilómetros, que luego se multiplica por la tasa correspondiente a la categoría asociada y la etiqueta del etiqueta ambiental.

### 6.1.3. Secciones

A continuación, presentamos una descripción más general de las secciones que delimitan los tramos predefinidos. Como se mencionó anteriormente, estas secciones son una representación aproximada de los tramos reales con sus puntos de control, ya que se han establecido de esta manera para fines de simulación. En el futuro, estos puntos podrán ser corregidos y ajustados de manera más precisa.

La figura que se muestra a continuación (Figura 6.1) representa la Sección AP-1. Esta sección está definida entre los puntos de referencia que se muestran en la Figura 6.2, que corresponde a la Sección AP-8.

la Sección AP-8 se delimita entre los puntos de referencia que se presentan en la Figura 6.3, que representa la Sección AP-68.

Esta estructura de imágenes y referencias visuales nos ayuda a entender cómo se dividen y relacionan estas diferentes secciones.

## 6. INTERFAZ DE PROCESAMIENTO DE DATOS - IPD (EAGLE)

---

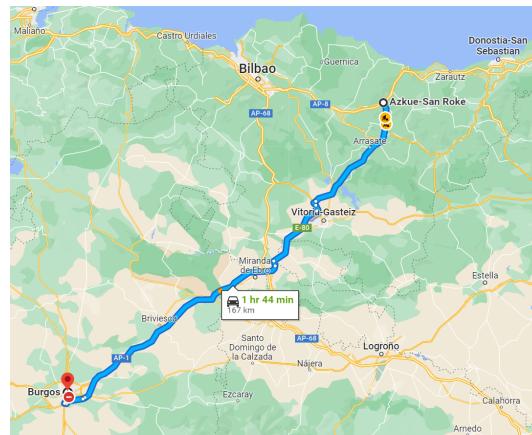


Figura 6.1: Sección AP-1

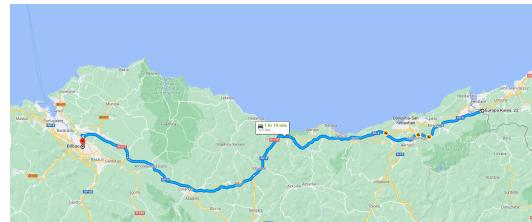


Figura 6.2: Sección AP-8

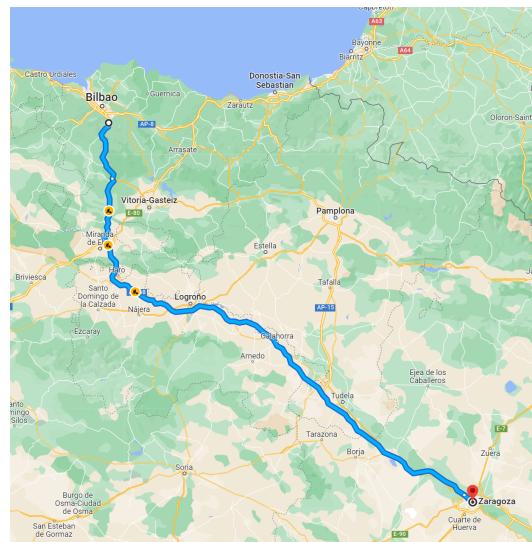
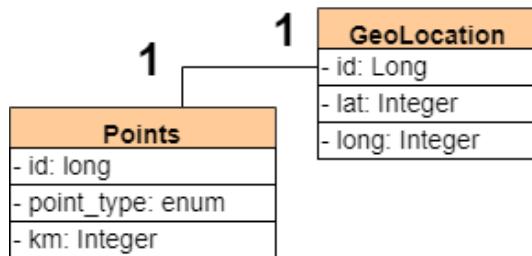


Figura 6.3: Sección AP-68

### 6.2. Incremento de Base de Datos I3

Durante el análisis del incremento 3 del módulo IPD, identificamos un cambio crítico en la estructura del modelo de dominio que resulta fundamental para calcular con precisión las transacciones de los usuarios. Este cambio implica la adición de la distancia en kilómetros de cada punto. Antes de esta modificación, no teníamos una metodología clara para calcular estas transacciones. La decisión de incorporar la distancia en kilómetros a cada punto se

basa en la necesidad de calcular las tasas en función de la distancia recorrida en el tramo, lo cual es el enfoque más apropiado. La alternativa era calcular el tiempo que se pasa en el tramo, pero esto requeriría también registrar la hora de entrada del vehículo. Por lo tanto, se optó por la primera opción. A continuación, se presentan los cambios realizados en el Modelo de Dominio (exclusivamente para el modelo de Puntos):



**Figura 6.4:** Cambio al Modelo de Dominio - Punto

Con esta actualización, se concluye nuestro análisis del módulo IPD.

### 6.3. Relación con los casos de uso IPD

En el capítulo 3 de Captura de requisitos se analizó los casos de uso del modulo IPD. Estos casos de uso son:

- **CU-0: Cobrar Usuario**
- **CU-1: Reconocer Matricula**
- **CU-2: Selección Tramo**

La lógica subyacente del módulo IPD se centra en el sistema de control de peajes, que opera de forma automática. Esto implica la gestión de la facturación de los usuarios, lo cual incluye la búsqueda de los usuarios mediante la matrícula en la base de datos, el reconocimiento de la matrícula a través de imágenes y el cálculo de las transacciones correspondientes.

El actor del sistema, el administrador, que se ha definido para este módulo, tiene como principal propósito facilitar la realización de estas demostraciones y la simulación del sistema.

En el sistema real, cada sección tendrá su propio servidor donde se ejecutarán estos casos de uso. En las siguientes secciones, se detallará con mayor profundidad la capa de servicios, cómo se implementa y programa este módulo. Por último, se llevará a cabo una demostración que incluirá la interfaz que se utilizará, la seguridad añadida, así como la funcionalidad y los datos asociados.

Se ha identificado otra entidad denominada "Cámaras", donde cada punto está asociado con un conjunto de cámaras. Sin embargo, esta entidad no se incluirá en el dominio, ya que no resulta necesaria para el funcionamiento actual del sistema y no desempeña un papel relevante en el mismo.

## 6. INTERFAZ DE PROCESAMIENTO DE DATOS - IPD (EAGLE)

Además de las modificaciones mencionadas, se ha identificado la necesidad de una funcionalidad crítica para el seguimiento de las secciones: la capacidad para determinar cuándo un usuario entra o sale de una sección específica. Con el sistema actual, esta funcionalidad no es factible de implementar de manera eficiente ni interpretarse adecuadamente. Por lo tanto, se ha decidido revisar la estructura del dominio de la siguiente manera:

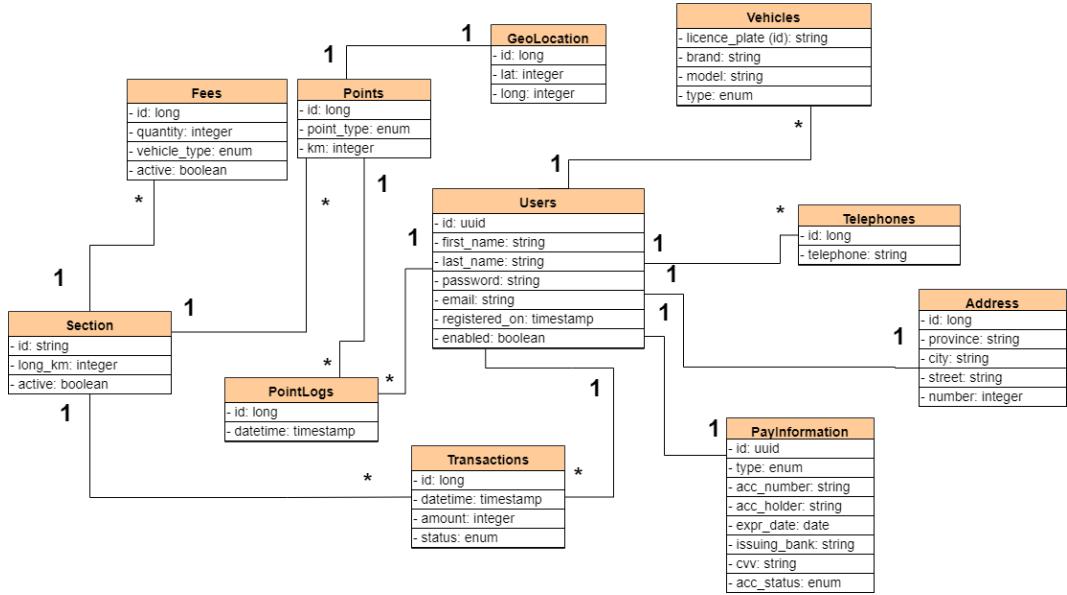


Figura 6.5: Diagrama de Dominio incremento 3

Se ha incorporado una nueva entidad denominada "PointLogs" que registra los eventos de acceso a cada sección. No es necesario establecer una conexión directa con las secciones, ya que cada punto está vinculado a una sección específica. Se presume que estos datos están almacenados de manera adecuada y que no existen puntos huérfanos sin una sección asociada. Con esta incorporación, hemos completado satisfactoriamente la configuración del tercer incremento de la base de datos.

### 6.4. Capa de Servicio de la IPD

En esta sección, se explicará la capa de servicios de la IPD, la cual integra el simulador de la IPD. Esta capa desempeña un papel fundamental al permitir la simulación del comportamiento del sistema de peaje, además de proporcionar servicios RESTful asociados y el reconocimiento de matrícula.

#### 6.4.1. Simulador IPD

La Interfaz de Procesamiento de Datos se centra en el hito principal del Incremento 3, que consiste en el simulador del sistema de peaje. Mientras que en el sistema real esta función se realiza de manera automática, en nuestro caso, necesitamos implementar esta simulación de forma manual. Este simulador está diseñado exclusivamente para el acceso de personal autorizado y se utiliza principalmente con fines de demostración y desarrollo.

La implementación del simulador involucra una serie de capas distintas y ofrece dos opciones de funcionamiento. Puede utilizar el reconocimiento de matrículas en su versión

2 a través de la API de Plate Recognizer, o puede optar por la versión 3 como se describe en el capítulo del incremento 5 (ver [8](#)). Para acceder al simulador, se utiliza el enlace específico proporcionado en la ruta /eagle.

#### 6.4.2. Servicios RESTful

El módulo IPD permite a los usuarios utilizar el servicio RESTful de las secciones y acceder a los registros de puntos K-anonimizados. Estas APIs RESTful se encuentran estructuradas de la siguiente manera:

- `api/v1/sections`: Esta API permite realizar consultas relacionadas con las secciones, tasas y ubicaciones.
- `api/v1/logs`: A través de esta API, los usuarios pueden realizar consultas sobre los registros de secciones específicas o generales, con datos que han sido anonimizados para proteger la privacidad de los individuos.

#### 6.4.3. Reconocimiento de Matrícula

El reconocimiento de matrículas se lleva a cabo a través de nuestro algoritmo desplegado en la versión 3 o mediante el uso de la versión 2 de la API de Plate Recognizer. La diferencia fundamental entre la versión 1 (v1) y la versión 2 (v2) de la API radica en la forma en que se manejan los datos.

En la versión 1 (v1), se requería la ruta absoluta de los archivos de imagen para realizar el reconocimiento de matrículas. Sin embargo, en un entorno real, esto se torna inviable debido a las restricciones de seguridad impuestas por los navegadores web, que prohíben el acceso a esta información sensible.

La versión 2 (v2) aborda este problema de manera efectiva al codificar la imagen en formato base64 y enviar una solicitud a la API de Plate Recognizer. Esta solución no solo supera las limitaciones de seguridad, sino que también mejora la interacción con el sistema, facilitando el proceso de reconocimiento de matrículas de manera más eficiente y segura.

### 6.5. Seguridad de la IPD

En esta sección, se proporciona una breve introducción a la seguridad en la IPD, ya que se aborda con mayor detalle en el Capítulo [5](#), titulado “Arquitectura del Sistema”.

El sistema Eagle se compone únicamente del simulador y los servicios RESTful, ambos incorporando medidas de seguridad. El acceso al simulador está restringido exclusivamente al personal de desarrollo y su propósito es simular el comportamiento del sistema.

Los servicios RESTful están protegidos mediante el uso de tokens JWT para garantizar su utilización segura. La API de Plate Recognizer proporciona un token de acceso a sus servicios REST, y nuestro sistema adopta una lógica similar para mejorar la eficacia y seguridad en el uso del sistema.



## CAPÍTULO

# 7

# Software de Visualización de Datos - SVD

En este capítulo, se presenta el Software de Visualización de Datos, la aplicación web. Aquí, se aborda la interfaz de usuario, los casos de uso previamente definidos en el Capítulo 3, Captura de Requisitos, y el flujo de la aplicación web.

## 7.1. Incremento de Base de Datos I3

Como se ha definido en la planificación, cada etapa del proceso de desarrollo implica un análisis detallado y la expansión de la Base de Datos. En el contexto del cuarto incremento, que se enfoca en el desarrollo de la aplicación web, no es una excepción esta expansión. En esta fase, hemos identificado un desafío significativo: la gestión y almacenamiento de la información del usuario, en particular, los datos de pago (PayInformation).

Para abordar este desafío, hemos tomado la decisión de gestionar esta información de manera externa, estableciendo una conexión con un servicio externo. Debido a limitaciones de tiempo, hemos implementado provisionalmente una entidad que indicará dónde se procesará el modo de pago. Como resultado de esta decisión, se realizarán modificaciones en el diagrama de dominio y en la estructura de la base de datos para reflejar este cambio.

En la Figura 7.1, se ha simplificado el diagrama eliminando la información de PayInformation, que no es relevante para el proyecto. Durante el análisis de esta etapa, ha surgido una pregunta importante: ¿Cómo puede el usuario o el sistema verificar que el usuario es el propietario legítimo de los vehículos? Existe la posibilidad de que otro usuario pueda agregar un vehículo con la misma matrícula.

Para abordar este tema, se han considerado dos medidas que no se llevarán a cabo funcionalmente en este proyecto, pero sí se implementarán a modo de demostración. La primera medida consiste en que el usuario suba sus documentos de identidad, ya sea el DNI o el NIE, y la segunda es la carga de los documentos del vehículo. Es fundamental recalcar que estas funcionalidades no formarán parte del sistema actual, sino que se incorporarán de forma demostrativa para ofrecer una representación completa del sistema. Es probable que estas dos cuestiones sean gestionadas por una entidad superior, como la DGT y el Gobierno.

## 7. SOFTWARE DE VISUALIZACIÓN DE DATOS - SVD

---

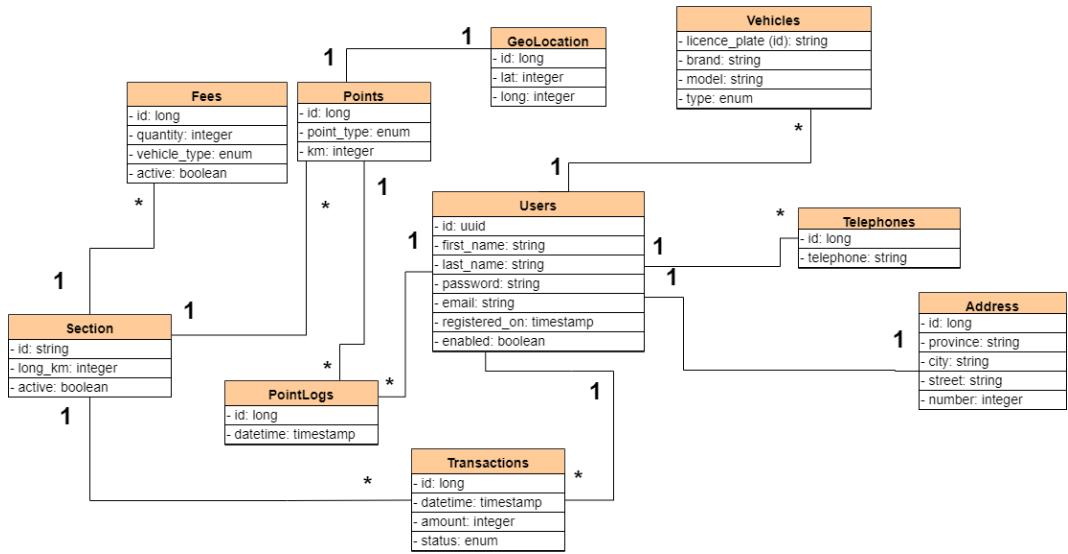


Figura 7.1: Diagrama de Dominio en el Incremento 4

Con esto, se asume que existen mecanismos externos de control y supervisión que verifican la autenticidad de los datos y la propiedad de los vehículos.

### 7.2. Relación con los Casos de Uso

En el capítulo 3 de Captura de requisitos se analizó los casos de uso del modulo SVD. Estos casos de uso son:

- **CU-4: Registro**
- **CU-5: Registro Vehículo**
- **CU-6: Confirmar Registro**
- **CU-7: Login/Autenticación**
- **CU-8: Gestionar Perfil**
- **CU-9: Ver Perfil**
- **CU-10: Modificar Perfil**
- **CU-11: Borrar Perfil**
- **CU-12: Logout**
- **CU-13: Gestión tramos**
- **CU-14: Ver tramos pendientes de pago**
- **CU-15: Seleccion tramo para pagar**
- **CU-16: Pagar**

- **CU-17: Ver tramos pagados**
- **CU-18: Ver tramos utilizados**
- **CU-19: Ver todos los tramos**

Durante esta etapa, el caso de uso 6 (CU-6) se considera opcional, ya que en el contexto del proyecto actual, no desempeña un papel fundamental. La idea detrás de este caso de uso es sencilla: enviar un correo electrónico al usuario con un código que debe verificar antes de acceder al sistema. Esto se implementa con el objetivo de reforzar la seguridad de la aplicación. Además de estos casos de uso, se ha introducido uno nuevo:

- **CU-20: Generar Token JWT para poder utilizar la API**

En el diseño presentado en la Figura 7.2, se destaca un componente importante: el API Docs o la referencia a la API del sistema. Este caso de uso permite a los usuarios generar su propio token JWT, lo que les habilita para utilizar los servicios REST del sistema de manera segura y eficiente.

## 7.3. Diseño de la aplicación Web

En esta fase, nos hemos centrado exclusivamente en la interfaz de usuario (UI) del proyecto. La experiencia del usuario (UX) es una metodología adicional que, lamentablemente, no ha sido factible de implementar debido a limitaciones de tiempo y recursos. Además, sería necesario contar con personal externo para realizar pruebas de opinión y analizar la experiencia del usuario de manera exhaustiva.

En cuanto al diseño, se ha optado por un estilo que incorpora colores verdes claros y un tema general de fondo blanco.



Figura 7.2: UI de la pagina inicial

La figura 7.2 representa la página de inicio de la aplicación, ofreciendo al usuario una visión del sistema futurista de peaje que se ha implementado bajo el nombre y el logotipo de .^aToll".

En cuanto a la elección de colores, hemos optado por el verde como color principal y el blanco como color secundario, siguiendo los principios de uso del color y el contraste que aprendí durante mi formación universitaria en una asignatura de UI/UX. El color verde, conocido por transmitir una sensación de tranquilidad, ha sido seleccionado con el objetivo de proporcionar a los usuarios una experiencia de uso segura y serena al interactuar con la aplicación web.

## 7.4. Seguridad

Se empleará una función hash previamente especificada, BCrypt, para garantizar el almacenamiento seguro de contraseñas. Además, se establecerá una cadena de filtros que restringirá el acceso a los enlaces, asegurando que los usuarios solo puedan acceder si están debidamente autenticados. La utilización de Spring Security nos brinda la flexibilidad necesaria para configurar una seguridad robusta y adaptable. La cadena de filtros se define de la siguiente manera:

**Listing 7.1:** Configuracion de Seguridad.

```
@Configuration
@EnableWebSecurity
@AllArgsConstructor
public class SecurityConfiguration {
    private final UserDetailsService userDetailsService;
    private final JwtRequestFilter jwtRequestFilter;
    private final AuthenticationEntryPoint jwtExceptionHandler;

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.addFilterBefore(jwtRequestFilter,
            UsernamePasswordAuthenticationFilter.class);
        http
            .csrf(httpSecurityCsrfConfigurer -> httpSecurityCsrfConfigurer
                .ignoringRequestMatchers("/authenticate", "/api/**"))
            .authorizeHttpRequests(authRegistry -> authRegistry
                .requestMatchers("/authenticate",
                    "/authenticate/**").permitAll()
                .requestMatchers("/api/**").authenticated()
                .requestMatchers("/js/**", "/style/**", "/font/**",
                    "/img/**", "/error").permitAll()
                .requestMatchers(
                    "/sections",
                    "/docs",
                    "/cookie-policy",
                    "/terms-privacy").permitAll()
                .requestMatchers(
                    "/",
                    "/about",
                    "/sign",
                    "/login").anonymous()
            .requestMatchers("/eagle", "/eagle/**").hasRole("ADMIN")
            .requestMatchers("/dashboard").hasRole("USER")
            .requestMatchers("/transactions").hasRole("USER")
    }
}
```

```

        .requestMatchers("/journeys").hasRole("USER")
        .requestMatchers("/profile", "/updateProfile",
                         "/deleteProfile").hasRole("USER")
        .anyRequest().authenticated() // Every other needs
                        authentication
    )
    .formLogin(login -> login
        .loginPage("/login")
        .defaultSuccessUrl("/dashboard", true)
        .failureUrl("/login?error=true")
        .usernameParameter("email")
    )
    .logout(logout -> logout
        .logoutUrl("/logout")
        .invalidateHttpSession(true)
        .deleteCookies("JSESSIONID")
        .logoutSuccessUrl("/")
    )
    .exceptionHandling(excep -> excep
        .accessDeniedPage("/error")
        .accessDeniedHandler(new CustomAccessDeniedHandler())
        .authenticationEntryPoint(jwtExceptionHandler)
    );
}

http.sessionManagement(sessm -> sessm
    .sessionFixation()
    .migrateSession()
    .maximumSessions(3)
);
return http.build();
}

@Bean
public DaoAuthenticationProvider authenticationProvider() {
    DaoAuthenticationProvider provider = new DaoAuthenticationProvider();
    provider.setPasswordEncoder(passwordEncoder());
    provider.setUserDetailsService(userDetailsService);
    return provider;
}

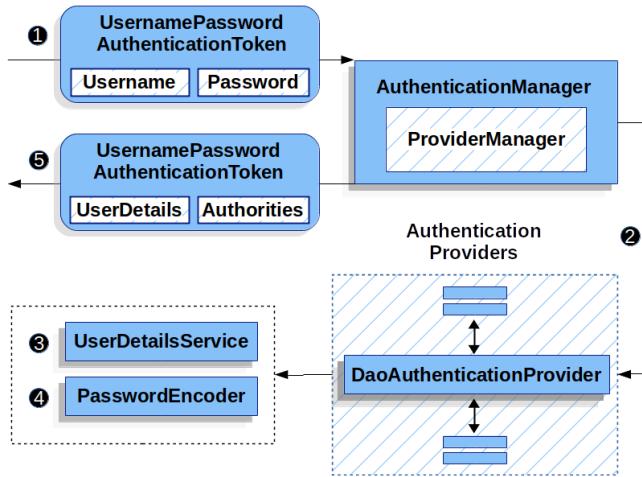
@Bean
public BCryptPasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}

```

#### 7.4.1. Autenticación de los usuarios

Estamos utilizando el proveedor DaoAuthentication de Spring Security, que cuenta con un flujo de autenticación ampliamente reconocido.

La figura que se muestra a continuación se ha obtenido de la página oficial de la



**Figura 7.3:** DaoAuthenticationProvider esquema

documentación de Spring Security<sup>1</sup>.

El flujo de autenticación:

1. Se crea un token de autenticación que contiene el nombre de usuario y la contraseña proporcionados por el usuario.
2. A continuación, este token se pasa al Authentication Manager, donde se asigna al DaoAuthenticationProvider.
3. En este punto, se inicializan los servicios de detalles del usuario, lo que permite al sistema obtener información adicional sobre el usuario que está tratando de autenticarse.
4. Luego, se procede a la inicialización del codificador de contraseña, en este caso, BCrypt. Este codificador es responsable de cifrar las contraseñas de los usuarios para su almacenamiento seguro y su posterior comparación durante la autenticación.
5. El último paso del proceso de autenticación consiste en verificar si los datos proporcionados son correctos. Para ello, se encripta la contraseña ingresada y se compara con la versión encriptada almacenada en la base de datos. Si coincide, se considera una autenticación exitosa. Además, en este punto, se asignan las autoridades (roles) al usuario, ya que los filtros y restricciones de acceso establecidos en la aplicación se basan en la seguridad basada en roles.

## 7.5. Elemento de búsqueda

Una de las características destacadas de la aplicación web es la barra de búsqueda, claramente visible en la figura 7.2. Esta barra tiene un propósito fundamental: facilitar la búsqueda de los endpoints definidos en la aplicación.

---

<sup>1</sup><https://docs.spring.io/spring-security/reference/servlet/authentication/passwords/dao-authentication-provider.html>

## 7.6. Captura de errores

Para lograr esta funcionalidad, hemos utilizado JavaScript junto con una estructura de datos altamente eficiente diseñada especialmente para autocompletado: el "árbol Trie".

Un árbol Trie es una estructura en la que cada letra se representa como un nodo, y cada letra tiene sus propios nodos hijos que combinan para formar palabras completas. Así se representa el árbol gráficamente:

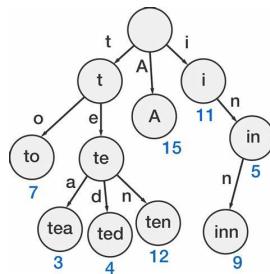


Figura 7.4: Arbol Trie

En nuestro caso, no hemos implementado todas las palabras posibles, sino únicamente aquellas que están relacionadas con los endpoints específicos de nuestro proyecto. Esto se hace para optimizar el rendimiento y la precisión de la búsqueda, asegurándonos de que los resultados estén directamente relacionados con las funciones y características de la aplicación que los usuarios están buscando. Esta estrategia permite una experiencia de usuario más eficiente y facilita la navegación en la aplicación web.

A continuación, se presenta la funcionalidad de esta barra de búsqueda:

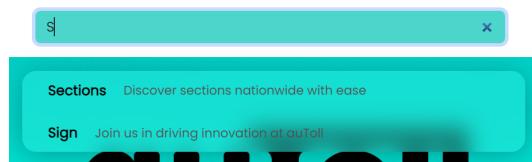


Figura 7.5: Barra de búsqueda 1



Figura 7.6: Barra de búsqueda 2

## 7.6. Captura de errores

Los errores que ocurran durante la ejecución de la aplicación se visualizan en la página de errores, a través del endpoint "/error". Spring, por defecto, establece este endpoint para gestionar los errores. Sin embargo, en nuestra aplicación, hemos desarrollado nuestro propio controlador de errores que extiende el controlador de errores proporcionado por Spring. Esta personalización nos brinda una mayor flexibilidad para definir la lógica que deseamos aplicar en respuesta a los errores.

## 7. SOFTWARE DE VISUALIZACIÓN DE DATOS - SVD

---

A continuación, se muestra un ejemplo de cómo manejamos un error 404, que indica que el contenido solicitado no ha sido encontrado:

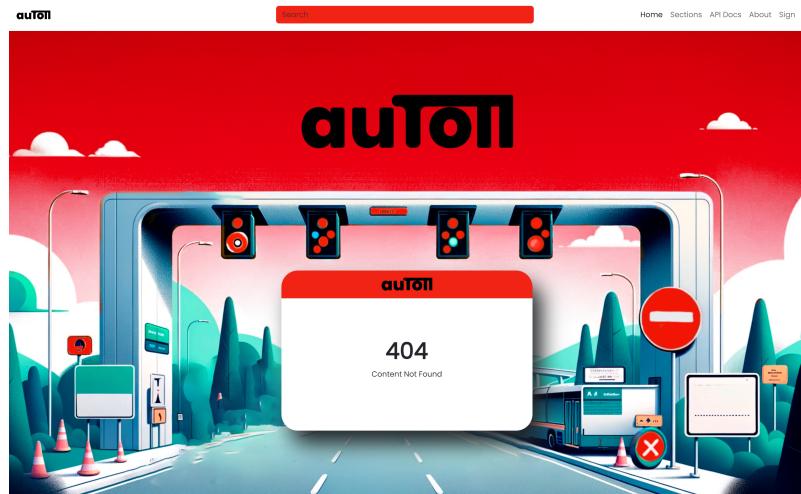


Figura 7.7: Pagina de error

### 7.7. Panel de control

El panel de control es la pestaña principal que los usuarios pueden acceder después de autenticarse en la aplicación. Los usuarios no autenticados serán redirigidos a la página de inicio de sesión. El panel de control ofrece tres opciones principales para que los usuarios utilicen la aplicación:

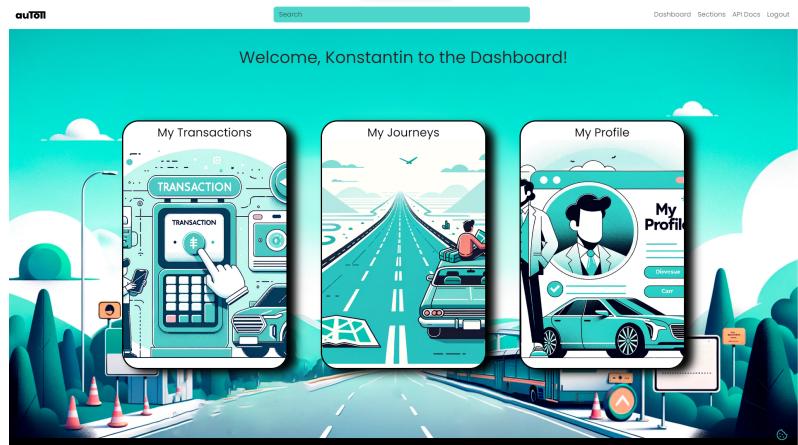


Figura 7.8: Panel de control

1. Transacciones: Donde los usuarios pueden ver todas sus transacciones
2. Viajes: Donde los usuarios pueden ver sus viajes realizados
3. Perfil de usuario: Donde los usuarios pueden, borrar, y modificar su cuenta.

## 7.8. Tramos

La pestaña 'Tramos' permite a todos los usuarios visualizar las autopistas en un mapa, siguiendo un enfoque similar al sitio web del proveedor actual de peajes de autopistas, ViaT. Sin embargo, nuestra aplicación destaca al generar dinámicamente las secciones en el mapa en lugar de hacerlo de forma manual, como se hace en la página de ViaT. A continuación, se muestra una imagen de cómo aparecen estas secciones en el mapa.

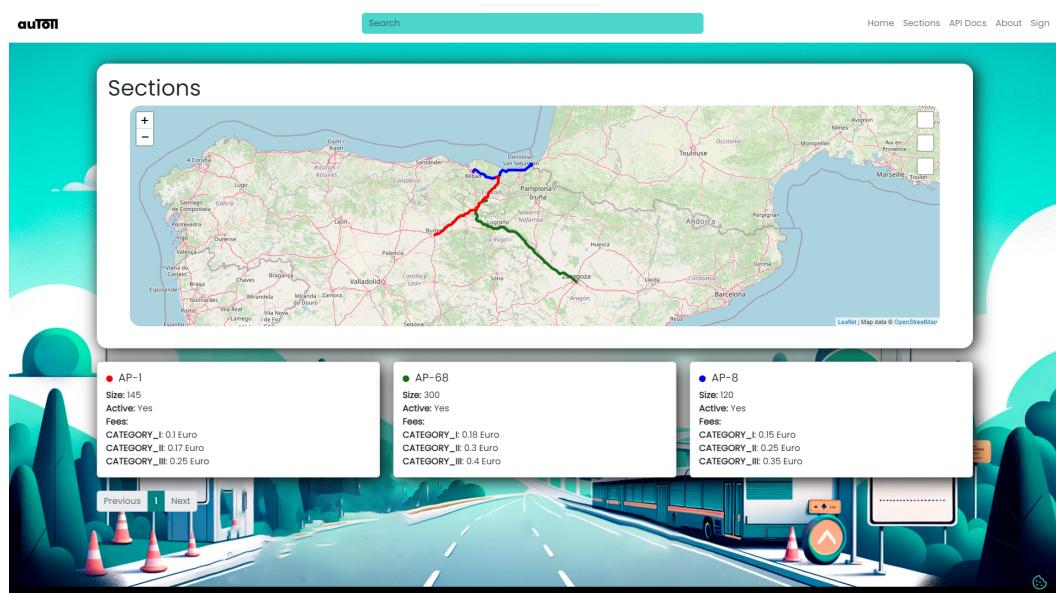


Figura 7.9: Tramos

Para lograr esta funcionalidad, aprovechamos la API de código abierto Leaflet<sup>2</sup> y su motor de generación de rutas, lo que nos brinda un enfoque dinámico. Cada tramo tiene puntos asignados, y utilizamos los puntos máximo y mínimo de cada tramo para generar automáticamente la línea que delimita dicho tramo en el mapa.

## 7.9. Generación JWT para nuestra API

Tal como se mencionó al inicio de este capítulo, se ha incorporado un nuevo caso de uso para la generación de tokens JWT destinados al uso de nuestra API. Esto es especialmente útil para desarrolladores que deseen utilizar nuestra API de secciones y viajes en sus propias aplicaciones u otros entornos.

Este proceso establece una autenticación para cada usuario, generando un token web que les permite acceder a la API. Gracias a este token, los usuarios pueden incorporar la API en sus aplicaciones y realizar consultas de manera efectiva. Sin este token, no podrán realizar ninguna consulta ni acceder a los recursos de la API.

<sup>2</sup><https://leafletjs.com/>



# 8

## CAPÍTULO

# Visión por Computadora

En este capítulo se detalla el incremento cinco, que se enfoca específicamente en el desarrollo de nuestro propio algoritmo y en la captura de las etiquetas ambientales de los vehículos.

## 8.1. Algoritmo de Reconocimiento

En esta sección, se detallarán los pasos del algoritmo de reconocimiento de matrículas en su fase “inicial”. Esta etapa es fundamental en el desarrollo de nuestro propio algoritmo. La decisión de crear nuestro sistema de reconocimiento de matrículas es bastante clara, ya que nos brinda un mayor control sobre los datos que manejamos, la forma en que los procesamos y su posterior uso.

### 8.1.1. OpenCV y Tesseract

En esta sección, se detallarán las cuatro etapas más importantes del algoritmo de reconocimiento de matrículas y se explicará el uso de las tecnologías OpenCV y Tesseract.

**OpenCV:** OpenCV (Open Source Computer Vision Library)<sup>1</sup> es una biblioteca de código abierto ampliamente utilizada en el campo de la visión por computadora y en procesamiento de imágenes. Ofrece una amplia gama de funciones y herramientas para realizar diversas tareas de procesamiento de imágenes y videos. En este algoritmo, OpenCV se utiliza para convertir la imagen en escala de grises, aplicar el filtro gaussiano, detectar bordes y contornos, y crear una máscara para el filtrado del contorno de la matrícula.

**Tesseract:** Tesseract<sup>2</sup> es un motor de OCR de código abierto desarrollado por Google. Es altamente preciso en la extracción de texto a partir de imágenes. Se utiliza para reconocer texto en las imágenes y convertir en texto legible por la máquina. En este algoritmo, Tesseract desempeña un papel fundamental en la extracción del texto de la matrícula después de que se haya detectado y confirmado la región de la matrícula.

---

<sup>1</sup><https://pypi.org/project/opencv-python/>

<sup>2</sup><https://github.com/tesseract-ocr/tesseract>

### 8.1.2. Etapas del Algoritmo de Reconocimiento de Matrículas

1. **Decodificación y Preprocesamiento de la Imagen:** En esta primera etapa, la imagen se decodifica a partir de una representación en base64. Luego, se realiza el preprocesamiento de la imagen utilizando la biblioteca OpenCV. Se convierte la imagen a escala de grises para simplificar el análisis y se aplica un filtro gaussiano para difuminar la imagen y reducir el ruido.
2. **Detección del Contorno de la Matrícula:** La segunda etapa implica la detección del contorno de la matrícula en la imagen preprocesada. Para ello, se utiliza el operador Canny de OpenCV para detectar bordes en la imagen. Luego, se encuentran los contornos en la imagen y se seleccionan los 40 contornos más grandes, ordenados por sus áreas. Esto se hace para tratar de identificar el contorno de la matrícula en la imagen.
3. **Filtrado del Contorno de la Matrícula:** En esta fase, se realiza un proceso de filtrado para confirmar que el contorno encontrado realmente corresponde a una matrícula. La idea es seleccionar el contorno cuyos valores de los píxeles se acerque más al blanco. Obviamente, esta aproximación solamente funcionará cuando el color del fondo de las matrículas sea blanca. Para ello, se ha creado una máscara para evaluar la intensidad de los píxeles dentro de este contorno seleccionado, así poder determinar si se asemeja a una matrícula o no. Si se cumple cierto umbral de intensidad, se considera que se ha encontrado la matrícula. Los países como Luxemburgo y Holanda utilizan matrículas con fondo amarillo [17], por ende, se ha tratado de encontrar un umbral válido tanto para matrículas de fondo blanco como amarillo.
4. **Extracción del Texto de la Matrícula:** La última etapa implica la extracción del texto de la matrícula. Se aísla la región de interés en escala de grises de la imagen original que corresponde a la matrícula. Luego, se utiliza Tesseract, para extraer el texto de esta región.

### 8.1.3. Demostración del Algoritmo

Aquí se muestran las etapas en forma de imágenes:

## 8.2. Captura de las Etiquetas Ambientales

Opcionalmente, se había contemplado la posibilidad de que el sistema pudiera capturar las etiquetas ambientales desde la propia imagen. Desafortunadamente, la DGT no ofrece una solución que supere la nuestra. Optamos por implementar la técnica de web scraping utilizando la herramienta Scrapy, que simplifica este proceso. El desafío principal radica en la necesidad de monitorear diariamente el sitio web de la DGT en busca de cambios o actualizaciones en el funcionamiento de este servicio.

### 8.2.1. Web Scraping con Scrapy

El web scraping con Scrapy es bastante sencillo.

## 8.2. Captura de las Etiquetas Ambientales



Figura 8.1: Ejecución del algoritmo

**Scrapy:** Scrapy<sup>3</sup> de código abierto desarrollado en Python se utiliza para realizar web scraping y web crawling de manera eficiente. Su objetivo principal es facilitar la extracción de datos de sitios web de forma estructurada y automatizada.

A continuación, se muestra un ejemplo de un spider Scrapy que se utiliza para capturar información de la web DGT:

Listing 8.1: Placement of the patch.

```
import scrapy
from scrapy.crawler import CrawlerProcess

class DGTSpider(scrapy.Spider):
    name = 'dgt_spider'

    def __init__(self, matricula=None, *args, **kwargs):
        super(DGTSpider, self).__init__(*args, **kwargs)
        self.matricula = matricula
        self.start_urls =
            [f'https://sede.dgt.gob.es/es/vehiculos/distintivo-ambiental/?accion=1&matriculahd=&matricula={self.matricula}&submit=Consultar#']

    def parse(self, response):
        distintivo =
            response.xpath('//div[@id="resultadoBusqueda"]/div/div/p/strong/text()').extract_first()
        print(distintivo)

if __name__ == "__main__":
    import sys
    process = CrawlerProcess()
    process.crawl(DGTSpider, matricula=sys.argv[1])
    process.start()
```

<sup>3</sup><https://scrapy.org/>

### 8.2.2. Resultados del Web Scraping

El resultado del scraping proporciona el texto extraído del XPath del HTML, que puede corresponder a uno de los siguientes casos:

1. Etiqueta Amarilla B
2. Etiqueta Verde C
3. Etiqueta ECO
4. Etiqueta 0
5. Sin etiqueta

Esta lista describe las posibles salidas que se pueden obtener al realizar el web scraping en el sitio web.

# 9

## CAPÍTULO

# Pruebas del Sistema

En este capítulo, se detallarán las pruebas efectuadas, incluyendo su tipo, así como la metodología empleada para evaluar el cumplimiento de los requisitos no funcionales definidos.

## 9.1. Pruebas de Servicios

Durante el proceso de desarrollo de los servicios, se llevaron a cabo dos tipos de pruebas ampliamente reconocidas y fundamentales para garantizar la funcionalidad y la integridad del software. Estos dos tipos de pruebas son:

### 9.1.1. Pruebas Unitarias

Las pruebas unitarias se enfocan en evaluar un servicio de forma aislada, sin la integración de otros componentes. En otras palabras, se simula el uso del servicio sin considerar la interacción con otros servicios. Estas pruebas se realizaron utilizando la herramienta Mockito. En efecto, se llevó a cabo la prueba exhaustiva de la capa de servicio de los componentes principales, tales como los relacionados con usuarios, tramos y tasas. Esto se hizo para garantizar que la lógica de negocio funcionara correctamente antes de avanzar con la implementación del incremento 4. De esta manera, aseguramos que la funcionalidad central esté en funcionamiento antes de proceder con el desarrollo de las interfaces frontales. A continuación, se proporciona un fragmento de código que muestra las pruebas realizadas:

**Listing 9.1:** Prueba unitaria

```
class FeeServiceImplTest {  
    @Mock  
    private FeeRepository feeRepository;  
    @Mock  
    private FeeDTOMapper feeDTOMapper;  
    @Mock  
    private SectionService sectionService;  
    @InjectMocks
```

## 9. PRUEBAS DEL SISTEMA

---

```
private FeeServiceImpl feeService;

@BeforeEach
void setUp() {
    MockitoAnnotations.openMocks(this);
}

@Test
void findAll_ReturnsListOfFeeDTO() {
    // Arrange
    List<Fee> fees = Arrays.asList(
        new Fee(0, VehicleType.CATEGORY_I),
        new Fee()
    );

    when(feeRepository.findAll(any(Pageable.class))).thenReturn(new
        PageImpl<>(fees));
    when(feeDTOMapper.apply(any(Fee.class))).thenReturn(
        new FeeDTO(
            0L,
            5,
            VehicleType.CATEGORY_I,
            Boolean.TRUE)
    );
}

// Act
List<FeeDTO> result = feeService.findAll();

// Assert
System.out.println(result);
verify(feeRepository, times(1)).findAll(any(Pageable.class));
assertThat(result).hasSize(feeds.size());
assertThat(result.get(0).vehicleType()).isEqualTo(VehicleType.CATEGORY_I);
}

.....

@Test
void update_ExistingId_ReturnsUpdatedFeeDTO() {
    // Arrange
    long existingId = 1L;
    FeeRequest feeRequest = new FeeRequest(3, VehicleType.CATEGORY_II,
        Boolean.FALSE);
    Fee existingFee = new Fee(existingId, 3, VehicleType.CATEGORY_I, null,
        Boolean.TRUE);
    Fee updatedFee = new Fee(
        existingId,
        feeRequest.quantity(),
        feeRequest.vehicleType(),
        null,
        feeRequest.active());
    FeeDTO expectedPointDTO = new FeeDTO(
        updatedFee.getFeeId(),
```

```

        updatedFee.getQuantity(),
        updatedFee.getVehicleType(),
        updatedFee.getActive());

when(feeRepository.findById(existingId)).thenReturn(Optional.of(existingFee));
when(feeRepository.save(any(Fee.class))).thenReturn(updatedFee);
when(feeDTOMapper.apply(updatedFee)).thenReturn(expectedPointDTO);

// Act
FeeDTO result = feeService.update(existingId, feeRequest);

// Assert
verify(feeRepository, times(1)).findById(existingId);
verify(feeRepository, times(1)).save(any(Fee.class));
assertThat(result.vehicleType()).isEqualTo(expectedPointDTO.vehicleType());
}

@Test
void update_NonExistingId_ThrowsException() {
    // Arrange
    long nonExistingId = 2L;
    FeeRequest feeRequest = new FeeRequest(0, VehicleType.CATEGORY_II,
        true);
    when(feeRepository.findById(nonExistingId)).thenReturn(Optional.empty());

    // Act & Assert
    assertThrows(FeeNotFoundException.class, () ->
        feeService.update(nonExistingId, feeRequest));
    verify(feeRepository).findById(nonExistingId);
    verify(feeRepository, never()).save(any(Fee.class));
}

```

### 9.1.2. Pruebas de Integración

Las pruebas de integración se centran en evaluar la interacción de un servicio con otros servicios en el contexto de la aplicación. Son esenciales para garantizar que todos los componentes funcionen de manera conjunta y que la interacción con la base de datos sea efectiva. Estas pruebas se llevaron a cabo específicamente en el módulo de Reconocimiento de Matrículas y su algoritmo, dado que este constituye una parte fundamental del sistema. En estas pruebas, se evalúa toda la interactividad, incluyendo la interacción con la base de datos. Una vez completadas las pruebas, se procede a eliminar los datos utilizados en la base de datos. Es importante destacar que, en el caso de las pruebas de integración, se simuló la conexión con la API de reconocimiento de matrículas versión 2, debido a las limitación previamente establecida en las llamadas a la API. A continuación se proporciona un fragmento de código:

**Listing 9.2:** Prueba de integracion

---

```

@SpringBootTest
@Transactional
@Slf4j
@AutoConfigureTestDatabase(replace = AutoConfigureTestDatabase.Replace.NONE)

```

## 9. PRUEBAS DEL SISTEMA

---

```
class EagleServiceIntegrationTest {
    private final EagleService eagleService;
    private final UserService userService;
    private final PointService pointService;
    private final FeeService feeService;
    private final VehicleService vehicleService;
    private final SectionService sectionService;

    @MockBean
    private SnapshotCloudAPI snapshotCloudAPI;
    private EagleRequest eagleRequest;

    @Autowired
    public EagleServiceIntegrationTest(EagleService eagleService, UserService
        userService, PointService pointService, FeeService feeService,
        VehicleService vehicleService, SectionService sectionService) {
        this.eagleService = eagleService;
        this.userService = userService;
        this.pointService = pointService;
        this.feeService = feeService;
        this.vehicleService = vehicleService;
        this.sectionService = sectionService;
    }

    @BeforeEach
    void setUp() {
        when(snapshotCloudAPI.recognizeVehicle(any(), any()))
            .thenReturn("5555ABC");

        // Add sections
        System.out.println("Saving sections");
        SectionRequest sectionRequest1 = new SectionRequest("AP-10", 80, true);
        SectionRequest sectionRequest2 = new SectionRequest("AP-80", 120, true);
        SectionRequest sectionRequest3 = new SectionRequest("AP-680", 300,
            true);
        .....
        sectionService.addPoint(section3, point2);
        sectionService.addPoint(section3, point3);
        sectionService.addPoint(section3, point4);
        sectionService.addPoint(section3, point5);
        sectionService.addPoint(section3, point6);
        sectionService.addPoint(section3, point7);
        sectionService.addPoint(section3, point8);
        System.out.println(sectionService.findAll());
    }

    @Test
    void detectVehicleTest() {
        System.out.println("=====|> ENTER TEST");

        eagleRequest = new EagleRequest(2L, "SOME_FILE", new
```

```
        Date(System.currentTimeMillis());
        System.out.println(eagleService.detectVehicle(eagleRequest));

        System.out.println("=====|> EXIT TEST");

        eagleRequest = new EagleRequest(18L, "SOME_FILE", new
            Date(System.currentTimeMillis()));
        System.out.println(eagleService.detectVehicle(eagleRequest));
    }
```

---

## 9.2. Métricas y Verificación de Requisitos

Como se estableció en el capítulo 3 sobre Captura de Requisitos, hemos definido varios requisitos no funcionales para garantizar el correcto funcionamiento del sistema.

### Rendimiento

1. **Optimización del procesamiento de datos:** Este requisito se evaluó midiendo el tiempo promedio de ejecución de los algoritmos v2 y v3 durante una transacción. Realizamos un total de 12 mediciones (3 repeticiones con 3 matrículas diferentes). Definimos una transacción como el período desde que el usuario inicia hasta que finaliza la operación. Los resultados son los siguientes:
  - **Algoritmo v2:** Tiempo promedio de 1726.33 milisegundos (~1.7 segundos) en el punto de inicio y 2926 milisegundos en el punto de finalización, ya que en este último se calcula y crea la transacción.
  - **Algoritmo v3:** Tiempo promedio de 1454 milisegundos (~1.5 segundos) en el punto de inicio y 2541 milisegundos en el punto de finalización.
2. **Objetivo:** Nuestro objetivo es lograr un promedio de menos de 2.5 segundos o 2500 milisegundos para todo el proceso de transacción.
3. **Resultado:** Se cumple el objetivo para los puntos de inicio, pero no para los puntos de finalización en ambos casos.

En el futuro, debemos revisar y optimizar los algoritmos de procesamiento de transacciones. Uno de los posibles problemas son la cantidad de consultas a la base de datos y la ejecución de procesos externos, como el script de Python de las etiquetas ambientales al realizar el web scraping. Observamos que al tener nuestro propio algoritmo, el tiempo de ejecución se reduce. Sin embargo, es importante destacar que nuestro algoritmo se encuentra en una fase inicial y tiene limitaciones. Además, estas pruebas se realizaron solo con matrículas centradas horizontalmente.

### Mantenibilidad

1. **Facilidad de mantenimiento:** Se realizó un análisis del tiempo planificado y continuo para el desarrollo de varias tareas, incluyendo el desarrollo del SVD, JWT, Pipeline, servicios y REST. Los resultados se desglosan de la siguiente manera:

## 9. PRUEBAS DEL SISTEMA

---

- **Objetivo:** Lograr un promedio de menos de una semana para el desarrollo o incorporación de nuevas funcionalidades.
  - **Resultado:** El promedio de tiempo para el desarrollo de estas funcionalidades es de 3.2 días, con la excepción del SVD, que requirió 6 días de desarrollo. Dado que el objetivo era completar estas tareas en 5 días o menos, se puede concluir que se cumple con lo establecido.
2. **Documentación:** Se documentaron las partes más críticas del sistema, específicamente los servicios que contienen la lógica de negocio, representando el 73 % de la aplicación. El objetivo era cubrir al menos el 80 % de la documentación. Por lo tanto, no se logró cumplir con este objetivo. Es necesario documentar los endpoints y las configuraciones para alcanzar esta meta.

### Tolerancia a Fallos

1. **Tasa de fallos en el reconocimiento de matrículas:** Se analizaron los dos algoritmos de reconocimiento, aunque solo el algoritmo v2 proporciona información sobre la precisión.
  - **Objetivo:** Obtener menos de 1 fallo en cada 10 intentos.
  - **Resultado:** El objetivo se cumple en ambos algoritmos, considerando que las matrículas utilizadas están centradas horizontalmente, como se supone que estarán dispuestas frente a las cámaras. Solo en el algoritmo v2 se pudo obtener la precisión del algoritmo, que promedia un 94 % de aciertos.

# 10 |

## CAPÍTULO

# Seguimiento y Control

En este capítulo se documentan las desviaciones significativas y los cambios en la planificación, así como los retrasos y las modificaciones en las decisiones del sistema.

## 10.1. Control de Cambios

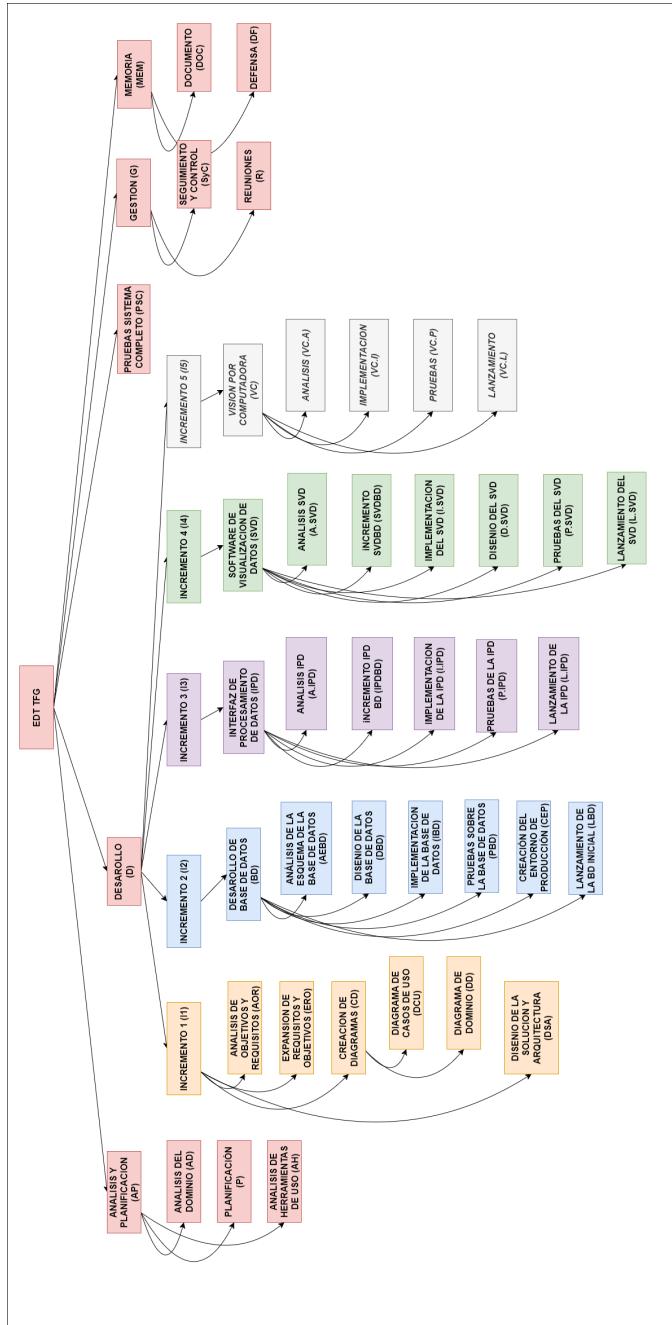
Durante la tercera reunión del proyecto AR-SyC-02, que tuvo lugar el 5 de septiembre de 2023 (ver acta [12.4](#)), se decidió postergar la implementación de la parte de visión por computadora hasta una etapa posterior. En consecuencia, se añadió un quinto incremento opcional, que abordara el análisis y la creación de un algoritmo propio, así como la expansión del reconocimiento de etiquetas ambientales en los vehículos durante el proceso de reconocimiento de matrículas. Esta modificación implicó una expansión en la lista de incrementos de la siguiente manera:

### ■ Incremento Visión por Computadora (VC):

- **Análisis:** En esta fase, se llevará a cabo un análisis exhaustivo de los algoritmos potenciales para la implementación de un algoritmo propio. Además, se evaluará la viabilidad del reconocimiento de etiquetas ambientales en los vehículos.
- **Implementación:** En esta etapa, se procederá a la implementación de todas las funcionalidades correspondientes a los nuevos algoritmos, siguiendo las decisiones tomadas durante la fase de análisis.
- **Pruebas:** Se llevarán a cabo pruebas exhaustivas que evaluarán tanto la precisión como el rendimiento del sistema de reconocimiento implementado.
- **Lanzamiento:** Gradualmente, las funcionalidades desarrolladas en este incremento se integrarán en el sistema actual, reemplazando progresivamente el sistema de reconocimiento de terceros previamente utilizado.

Con esta modificación, la Estructura de Desglose de Trabajo (EDT) incluye un nuevo paquete de trabajo en la sección correspondiente al Desarrollo (D) [10.1](#)

## 10. SEGUIMIENTO Y CONTROL



**Figura 10.1:** Estructura de Desglose de Trabajo (EDT) expandido

### ■ Incremento 5 (I5):

- Paquete de Trabajo de Visión por Computadora (VC): Se centra en el desarrollo de nuestro propio algoritmo de reconocimiento de matrículas, así como en la implementación del reconocimiento de etiquetas ambientales.

Además, dado que este incremento es opcional y su tiempo de desarrollo no puede estimarse con precisión en este momento, se decidió mantener sin cambios el modelo Gantt

existente.

## 10.2. Retrasos significativos

En esta sección se destacan y documentan los retrasos significativos que han sido identificados durante el seguimiento y control del proyecto.

### 10.2.1. Identificación de Retrasos

En esta subsección se identifican los retrasos significativos utilizando un código de identificación único:

#### 10.2.1.1. RS-1: Retraso en los incrementos 1 y 2

El primer retraso, identificado durante la tercera reunión [12.4](#) del proyecto AR-SyC-02 el 5 de septiembre de 2023 (ver acta [12.4](#)), se relaciona con la ejecución de tareas correspondientes a los incrementos 1 y 2. En el análisis se constató que aún no se habían iniciado las tareas del incremento 1, y se decidió que es necesario comenzar con las tareas del incremento 2, que implica el trabajo en la base de datos.

### 10.2.2. Causas de los retrasos

En esta subsección se identifican las causas de los retrasos:

#### 10.2.2.1. RS-1

Las causas de este retraso se debieron a que el enfoque del trabajo se centró en el desarrollo y prueba de la Interfaz de Procesamiento de Datos (IPD), especialmente en el algoritmo de reconocimiento de matrículas. Esto llevó a la decisión de establecer un nuevo incremento de Visión por Computadora (VC) para abordar específicamente este aspecto del proyecto, que implementaría nuestro propio algoritmo de reconocimiento y lectura de las etiquetas ambientales.

### 10.2.3. Impacto de los retrasos

En esta sub sección se analizan los impactos de los retrasos significativos.

#### 10.2.3.1. RS-1

A pesar del retraso identificado, su impacto ha sido relativamente limitado. Como medida de respuesta, se decidió acelerar el trabajo y se esperaba completar el incremento 1 dentro del tiempo previsto antes de la próxima reunión del proyecto. Se tomaron acciones adicionales para asegurar que el cronograma se mantenga en línea con las expectativas.

En el diagrama de Gantt, se había previsto originalmente que el incremento 1 se finalizara hasta la segunda semana de septiembre. Sin embargo, debido al retraso identificado, se implementó un esfuerzo adicional para acelerar este trabajo y asegurar su finalización dentro del plazo previsto. Esto conlleva a un aumento en las horas de dedicación para el primer incremento.

## 10. SEGUIMIENTO Y CONTROL

---

### 10.2.4. Acciones correctivas

#### 10.2.4.1. RS-1

Se llevó a cabo un esfuerzo adicional de trabajo acelerado con el objetivo de garantizar que el incremento 1 se completara dentro del plazo previsto.

Paquete de Trabajo	Tarea	Estimación
Paquete de trabajo Desarrollo-Incremento 1	I1.1 Análisis de Objetivos y Requisitos (AOR)	5h
	I1.2 Expansión de los Requisitos y Objetivos (ERO)	5h
	I1.3 Creación de Diagramas (CD)	20h
	I1.4 Diseño de la Solución y Arquitectura del sistema (DSA)	10h
	<b>SUBTOTAL I1</b>	<b>40h</b>
<b>HORAS TOTALES</b>		<b>347h</b>

**Tabla 10.1:** Tabla de acciones correctivas sobre rs-1

---

Como se puede observar en la tabla 10.1 esto resultó en un aumento tanto en las horas totales como en las tareas dentro del paquete de trabajo incremento 1 (I1).

Además, dentro del paquete de trabajo I3 relacionado con el módulo de la Interfaz de Procesamiento de Datos (IPD), la tarea IPD.1 se reubicó en el incremento 5, ya que se centró en el análisis de los algoritmos para el reconocimiento de matrículas.

Paquete de Trabajo	Tarea	Estimación
Paquete de trabajo Desarrollo- Incremento 3-Interfaz de Procesamiento de Datos (IPD)	IPD.1 Incremento necesario de la BD	5h
	IPD.2 Implementación de la interfaz.	20h
	IPD.3 Pruebas	15h
	IPD.4 Integración y entrega del IPD.	5h
	<b>SUBTOTAL IPD</b>	<b>45h</b>
<b>HORAS TOTALES</b>		<b>342h</b>

**Tabla 10.2:** Tabla de estimación de dedicación de las tareas

---

Esto implicó un cambio en los nombres de las tareas y una reducción en las horas asignadas al paquete de trabajo.

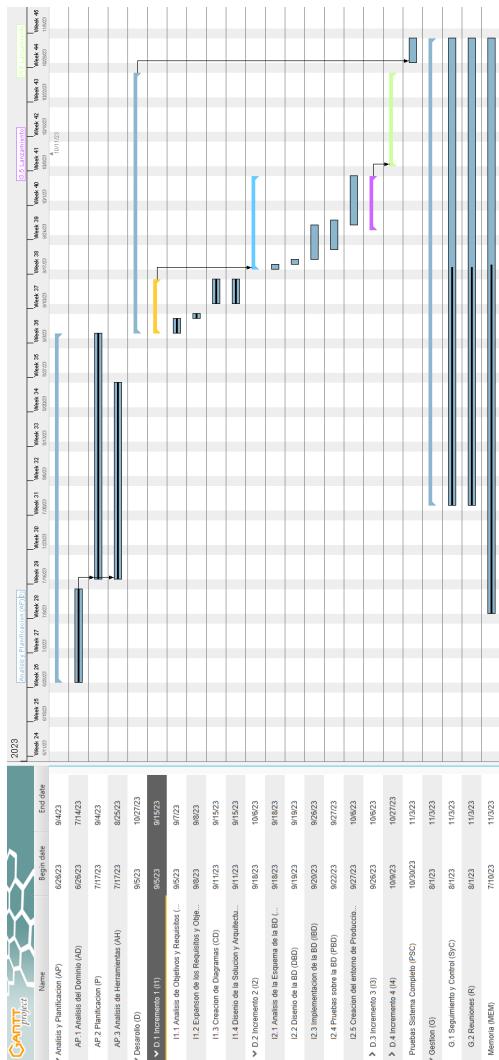
## 10.3. Replanificación del proyecto

En esta sección se aborda la replanificación del proyecto a medida que este avanza, incluyendo la identificación de las razones que la motivaron y su justificación.

### 10.3.1. Replanificación 1

La primera replanificación se tomó durante la cuarta reunión [12.4](#) en base a una observación conjunta realizada por el alumno y los tutores. Fue evidente que era necesario adelantar uno de los incrementos para poder llevar a cabo la defensa del proyecto en noviembre. Como resultado, se decidió reorganizar el plan, adelantando el incremento 3 y acelerando el trabajo en el incremento 2.

Esto implica realizar modificaciones en las estimaciones de horas de trabajo y en el diagrama de Gantt que se habían sido establecidos inicialmente en la planificación del proyecto.



**Figura 10.2:** Diagrama Gantt modificado, replanificación 1

## 10. SEGUIMIENTO Y CONTROL

---

Como se puede observar en la figura 10.2, se llevó a cabo una replanificación que afectó a los incrementos 2 y 3, resultando en un aumento en las horas de trabajo y solapamiento de tareas. Anteriormente, se había contemplado iniciar el incremento 3 una vez finalizado el 2; sin embargo, como se especificó en la reunión 12.4, se decidió acelerar el trabajo. Esta decisión no tuvo un impacto negativo en la ejecución de las tareas, pero implicó una coordinación más eficiente y sostenible en la integración con la base de datos.

Esta estrategia se pudo llevar a cabo, como se mencionó en la reunión, utilizando Hibernate, lo que simplificó en gran medida los cambios y la mantenibilidad. La capacidad de construir automáticamente la base de datos con Hibernate agilizó significativamente el proceso de desarrollo y redujo los impactos negativos.

Horas impactadas después de la replanificación:

Paquete de Trabajo	Tarea	Estimación
Paquete de trabajo Desarrollo-Incremento 2-Base de Datos (BD)	BD.1 Análisis de la Esquema de la Base de Datos (AEBD)	5h
	BD.2 Diseño de la Base de Datos (DBD)	15h
	BD.3 Implementación de la Base de Datos (IBD)	10h
	BD.4 Pruebas sobre la Base de Datos (PBD)	10h
	BD.5 Creación del Entorno de Producción (CEP)	6h
	BD.6 Lanzamiento de la Base de Datos Inicial (LBD)	5h
	<b>SUBTOTAL BD</b>	<b>51h</b>
Paquete de trabajo Desarrollo- Incremento 3-Interfaz de Procesamiento de Datos (IPD)	IPD.1 Incremento necesario de la BD	5h
	IPD.2 Implementación de la interfaz.	30h
	IPD.3 Pruebas	15h
	IPD.4 Integración y entrega del IPD.	10h
	<b>SUBTOTAL IPD</b>	<b>60h</b>
	<b>HORAS TOTALES</b>	<b>372h</b>

**Tabla 10.3:** Tabla de estimación de dedicación de las tareas

---

Estas modificaciones resultaron en un incremento en el total de horas de trabajo, aumentando de 342 a 372 horas en total.

### 10.3.2. Replanificación 2

La segunda replanificación se llevó a cabo después de la reunión seis (consultar 12.4). Tras esta reunión y el análisis de riesgos, se tomó la decisión de replanificar la fase final del proyecto, que corresponde al incremento 4. En esta decisión, se optó por no utilizar un framework dedicado como React, que se había considerado previamente en el capítulo 4 durante el análisis de herramientas. Sin embargo, se decidió incorporar el incremento 5, que originalmente era opcional, para completar integralmente el sistema.

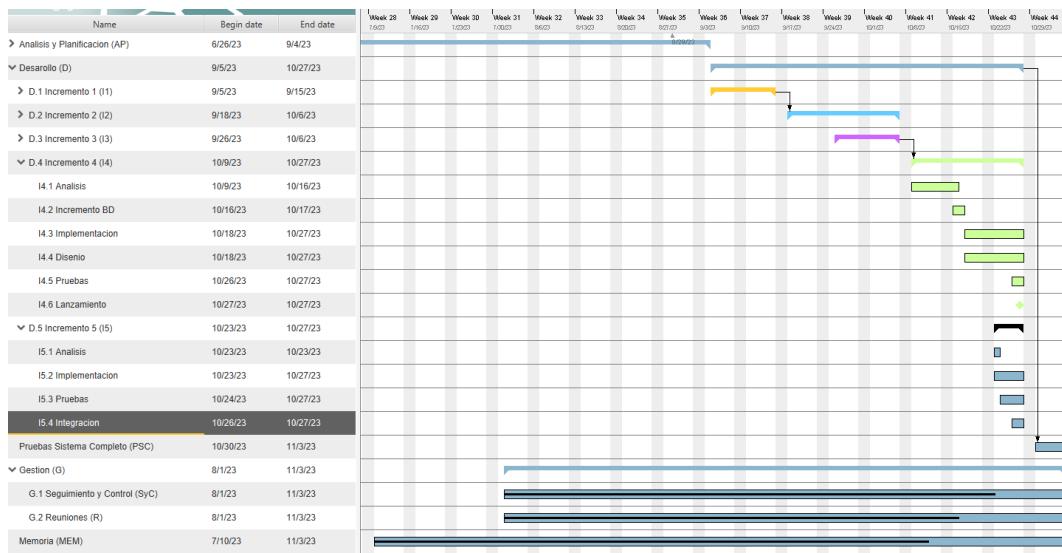
Esto implicó la utilización de la tabla de análisis de horas de la siguiente manera:

Paquete de Trabajo	Tarea	Estimación en horas (h)	Real en horas (h)
Paquete de trabajo Análisis y Planificación (AP)	AP.1 Análisis del Dominio (AD)	4	2
	AP.2 Planificación (P)	25	25
	AP.3 Análisis de las Herramientas (AH)	5	3
	<b>SUBTOTAL AP</b>	<b>34</b>	<b>30</b>
Paquete de trabajo Desarrollo-Incremento 1	I1.1 Análisis de Objetivos y Requisitos (AOR)	5	5
	I1.2 Expansión de los Requisitos y Objetivos (ERO)	5	3
	I1.3 Creación de Diagramas (CD)	20	20
	I1.4 Diseño de la Solución y Arquitectura del sistema (DSA)	10	5
	<b>SUBTOTAL I1</b>	<b>40</b>	<b>33</b>
Paquete de trabajo Desarrollo-Incremento 2-Base de Datos (BD)	BD.1 Análisis de la Esquema de la Base de Datos (AEBD)	5	4
	BD.2 Diseño de la Base de Datos (DBD)	15	15
	BD.3 Implementación de la Base de Datos (IBD)	10	8
	BD.4 Pruebas sobre la Base de Datos (PBD)	10	8
	BD.5 Creación del Entorno de Producción (CEP)	6	3
	BD.6 Lanzamiento de la Base de Datos Inicial (LBD)	5	3
	<b>SUBTOTAL BD</b>	<b>51</b>	<b>41</b>
Paquete de trabajo Desarrollo-Incremento 3-Interfaz de Procesamiento de Datos (IPD)	IPD.1 Incremento necesario de la BD	5	3
	IPD.2 Implementación de la interfaz.	30	31
	IPD.3 Pruebas	15	26
	IPD.4 Integración y entrega del IPD.	10	10
	<b>SUBTOTAL IPD</b>	<b>60</b>	<b>70</b>
Paquete de trabajo Desarrollo-Incremento 4-Software de Visualización de Datos (SVD)	SVD.1 Análisis de los requisitos para el SVD.	5	2
	SVD.2 Incremento necesario de la BD	5	2
	SVD.3 Implementación del SVD	20	0
	SVD.4 Diseño de la aplicación	20	0
	SVD.5 Pruebas.	10	0
	SVD.6 Integración y entrega del SVD.	2	0
	<b>SUBTOTAL SVD</b>	<b>62</b>	<b>4</b>
Paquete de trabajo Desarrollo-Incremento 5-Vision por Computadora	VC.1 Análisis de algoritmos y herramientas.	3	0
	VC.2 Implementación del VC	8	0
	VC.3 Pruebas.	2	0
	VC.4 Integración y entrega del VC.	2	0
	<b>SUBTOTAL VC</b>	<b>15</b>	<b>0</b>
Paquete de trabajo Pruebas Sistema Completo	<b>SUBTOTAL PSC</b>	<b>20</b>	<b>0</b>
Paquete de trabajo Gestión de Proyecto (GP)	GP.1 Seguimiento y Control (SyC)	10	9
	GP.2 Reuniones.	8	6
	<b>SUBTOTAL GP</b>	<b>18</b>	<b>15</b>
Paquete de trabajo Memoria (MEM)	MEM.1 Recopilación de información y datos relevantes y documentación	80	73
	MEM.2 Preparación para la defensa final del proyecto	7	0
	<b>SUBTOTAL MEM</b>	<b>87</b>	<b>73</b>
<b>HORAS TOTALES</b>		<b>387</b>	<b>266</b>
<b>LIMITE HORAS TOTALES</b>		<b>360</b>	
		<b>SOBRESALE</b>	<b>DENTRO DEL LIMITE</b>
		<b>27</b>	<b>0</b>

Figura 10.3: Análisis de horas en la replanificación 2

y los cambios en el diagrama de Gantt quedaron de la siguiente manera:

## 10. SEGUIMIENTO Y CONTROL



**Figura 10.4:** Diagrama Gantt en la replanificación 2

En términos generales, es importante destacar que no se produjeron cambios significativos, pero sí se experimentó un aumento considerable en la carga de trabajo requerida para alcanzar los hitos y plazos establecidos. Específicamente, este aumento se evidenció en un excedente de tiempo planificado de 27 horas debido a la incorporación del incremento 5.

En el proceso de replanificación, nos quedan tres semanas para cumplir con los plazos del incremento 4 y seguir la planificación original. No obstante, es importante tener en cuenta que si en el último momento se determina que el incremento 5 no es factible, se eliminará como una opción viable y no se tendrá en cuenta en el proyecto final.

Las horas del incremento 5 se establecieron de la siguiente forma:

Paquete de Trabajo	Tarea	Estimación
Paquete de trabajo Desarrollo- Incremento 3-Interfaz de Procesamiento de Datos (IPD)	VC.1 Análisis de algoritmo	3h
	VC.2 Implementación.	8h
	VC.3 Pruebas	2h
	VC.4 Integración y entrega del VC.	2h
	<b>SUBTOTAL VC</b>	<b>15h</b>
	<b>HORAS TOTALES</b>	<b>387h</b>

**Tabla 10.4:** Tabla de estimación de dedicación de las tareas incremento 5

### 10.4. Seguimiento de las horas trabajo

En esta sección se llevó a cabo un análisis de las horas de trabajo.

Como se mencionó durante la reunión 12.4, anteriormente las horas de trabajo no se estaban siguiendo de manera estricta, lo que dificultaba la estimación y el seguimiento realista del trabajo. Para abordar este problema, se implementó una hoja de cálculo en Excel dedicada al seguimiento de las horas de trabajo. Estas horas se registraron de manera específica por paquete de trabajo y se actualizaron a diario, lo que permitió obtener una visión precisa de la ejecución real del trabajo y detectar posibles desviaciones con respecto a las horas planificadas.

Para las tareas cuyas horas de trabajo no se habían seguido de esta manera, se realizaron

## 10.4. Seguimiento de las horas trabajo

### 10.4.1.1. Primer análisis de las horas de trabajo

Este análisis se enfocó en las horas de trabajo empleadas hasta alcanzar el hito de inicio del incremento 2. Este enfoque proporcionó una visión detallada de lo que se había planeado hasta ese momento y de lo que realmente se había empleado. Como se mencionó previamente, es importante destacar que las horas estimadas hasta el inicio del incremento 2 no eran precisas, por lo que en la imagen 10.5, que se presentó a continuación, estas horas se destacaron en color amarillo para resaltar esta discrepancia.

Paquete de Trabajo	Tarea	Estimación en horas (h)	Real en horas (h)
Paquete de trabajo Análisis y Planificación (AP)	AP.1 Análisis del Dominio (AD)	4	2
	AP.2 Planificación (P)	25	25
	AP.3 Análisis de las Herramientas (AH)	5	3
	<b>SUBTOTAL AP</b>	<b>34</b>	<b>30</b>
Paquete de trabajo Desarrollo-Incremento 1	I1.1 Análisis de Objetivos y Requisitos (AOR)	5	5
	I1.2 Expansión de los Requisitos y Objetivos (ERO)	5	3
	I1.3 Creación de Diagramas (CD)	20	20
	I1.4 Diseño de la Solución y Arquitectura del sistema (DSA)	10	5
	<b>SUBTOTAL I1</b>	<b>40</b>	<b>33</b>
Paquete de trabajo Desarrollo-Incremento 2- Base de Datos (BD)	BD.1 Análisis de la Esquema de la Base de Datos (AEBD)	5	4
	BD.2 Diseño de la Base de Datos (DBD)	15	10
	BD.3 Implementación de la Base de Datos (IBD)	10	2
	BD.4 Pruebas sobre la Base de Datos (PBD)	10	0
	BD.5 Creación del Entorno de Producción (CEP)	6	0
	BD.6 Lanzamiento de la Base de Datos Inicial (LBD)	5	0
Paquete de trabajo Desarrollo-Incremento 3- Interfaz de Procesamiento de Datos (IPD)	<b>SUBTOTAL BD</b>	<b>51</b>	<b>16</b>
	IPD.1 Incremento necesario de la BD	5	0
	IPD.2 Implementación de la interfaz.	30	0
	IPD.3 Pruebas	15	0
	IPD.4 Integración y entrega del IPD.	10	0
Paquete de trabajo Desarrollo-Incremento 4- Software de Visualización de Datos (SVD)	<b>SUBTOTAL IPD</b>	<b>60</b>	<b>0</b>
	SVD.1 Análisis de los requisitos para el SVD.	5	0
	SVD.2 Incremento necesario de la BD	5	0
	SVD.3 Implementación del SVD	20	0
	SVD.4 Diseño de la aplicación	20	0
	SVD.5 Pruebas.	10	0
Paquete de trabajo Pruebas Sistema Completo (PSC)	SVD.6 Integración y entrega del SVD.	2	0
	<b>SUBTOTAL SVD</b>	<b>62</b>	<b>0</b>
	<b>SUBTOTAL PSC</b>	<b>20</b>	<b>0</b>
Paquete de trabajo Gestión de Proyecto (GP)	GP.1 Seguimiento y Control (SyC)	10	5
	GP.2 Reuniones.	8	4
	<b>SUBTOTAL GP</b>	<b>18</b>	<b>9</b>
Paquete de trabajo Memoria (MEM)	MEM.1 Recopilación de información y datos relevantes y docu	80	55
	MEM.2 Preparación para la defensa final del proyecto	7	0
	<b>SUBTOTAL MEM</b>	<b>87</b>	<b>55</b>
<b>HORAS TOTALES</b>		<b>372</b>	<b>143</b>
<b>LÍMITE HORAS TOTALES</b>			
		<b>SOBRESEAL</b>	<b>DENTRO DEL LÍMITE</b>
		12	0

**Figura 10.5:** Primer análisis de las horas de trabajo empleadas

Las tareas resaltadas en verde indicaron que se realizaron dentro del plazo establecido, mientras que las tareas en rojo señalaron que tomaron más tiempo de lo estimado originalmente. La gráfica se segmentó en función de las tareas correspondientes a cada paquete de trabajo, con sus horas estimadas y las horas reales efectivamente empleadas. Además, se incluyó un análisis de las horas totales en caso de exceder el plan previsto.

En el caso de las horas planificadas, si se superaban las 360 horas estimadas, esto representaba un desfase de 12 horas con respecto al plan actual. El límite para las horas reales era igual al total de horas estimadas, lo que indicaba que aún se encontraba dentro

## 10. SEGUIMIENTO Y CONTROL

---

del margen de trabajo planificado.

Estas reflexiones nos permitieron identificar en qué tareas se había invertido más tiempo del previsto y cuáles habían sido sobrevaloradas en cuanto a su tiempo de ejecución.

### 10.4.1.2. Segundo análisis de las horas de trabajo

Este análisis se enfocó en las horas de trabajo empleadas hasta alcanzar el hito de inicio del último incremento. Este enfoque proporcionó una visión detallada de lo que se había planeado hasta ese momento y de lo que realmente se había empleado. A continuación se proporciona la tabla de hoja de cálculo con las horas:

Paquete de Trabajo	Tarea	Estimación en horas (h)	Real en horas (h)
Paquete de trabajo Análisis y Planificación (AP)	AP.1 Análisis del Dominio (AD)	4	2
	AP.2 Planificación (P)	25	25
	AP.3 Análisis de las Herramientas (AH)	5	3
	<b>SUBTOTAL AP</b>	<b>34</b>	<b>30</b>
Paquete de trabajo Desarrollo-Incremento 1	I1.1 Análisis de Objetivos y Requisitos (AOR)	5	5
	I1.2 Expansión de los Requisitos y Objetivos (ERO)	5	3
	I1.3 Creación de Diagramas (CD)	20	20
	I1.4 Diseño de la Solución y Arquitectura del sistema (DSA)	10	5
	<b>SUBTOTAL I1</b>	<b>40</b>	<b>33</b>
Paquete de trabajo Desarrollo-Incremento 2- Base de Datos (BD)	BD.1 Análisis de la Esquema de la Base de Datos (AEBD)	5	4
	BD.2 Diseño de la Base de Datos (DBD)	15	15
	BD.3 Implementación de la Base de Datos (IBD)	10	8
	BD.4 Pruebas sobre la Base de Datos (PBD)	10	8
	BD.5 Creación del Entorno de Producción (CEP)	6	3
	BD.6 Lanzamiento de la Base de Datos Inicial (LBD)	5	3
	<b>SUBTOTAL BD</b>	<b>51</b>	<b>41</b>
Paquete de trabajo Desarrollo- Incremento 3- Interfaz de Procesamiento de Datos (IPD)	IPD.1 Incremento necesario de la BD	5	3
	IPD.2 Implementación de la interfaz.	30	31
	IPD.3 Pruebas	15	26
	IPD.4 Integración y entrega del IPD.	10	10
	<b>SUBTOTAL IPD</b>	<b>60</b>	<b>70</b>
Paquete de trabajo Desarrollo- Incremento 4- Software de Visualización de Datos (SVD)	SVD.1 Análisis de los requisitos para el SVD.	5	2
	SVD.2 Incremento necesario de la BD	5	2
	SVD.3 Implementación del SVD	20	0
	SVD.4 Diseño de la aplicación	20	0
	SVD.5 Pruebas.	10	0
	SVD.6 Integración y entrega del SVD.	2	0
	<b>SUBTOTAL SVD</b>	<b>62</b>	<b>4</b>
Paquete de trabajo Pruebas Sistema Completo	<b>SUBTOTAL PSC</b>	<b>20</b>	<b>0</b>
Paquete de trabajo Gestión de Proyecto (GP)	GP.1 Seguimiento y Control (SyC)	10	9
	GP.2 Reuniones.	8	6
	<b>SUBTOTAL GP</b>	<b>18</b>	<b>15</b>
Paquete de trabajo Memoria (MEM)	MEM.1 Recopilación de información y datos relevantes y docu	80	73
	MEM.2 Preparación para la defensa final del proyecto	7	0
	<b>SUBTOTAL MEM</b>	<b>87</b>	<b>73</b>
<b>HORAS TOTALES</b>		<b>372</b>	<b>266</b>
<b>LÍMITE HORAS TOTALES</b>		<b>360</b>	<b>SOBRESALE DENTRO DEL LÍMITE</b>
		12	0

Figura 10.6: Segundo análisis de las horas de trabajo empleadas

Las tareas marcadas en verde, como se identificó en el análisis anterior, se completaron dentro del plazo previsto. En el incremento IPD, nos encontramos con varias tareas resaltadas en rojo debido a las pruebas exhaustivas, la expansión de la base de datos en esta etapa y el desarrollo de los servicios REST. Además, enfrentamos dificultades en la realización del pipeline de DevOps, lo que provocó un retraso en estas etapas en comparación con lo que se había planificado. Por otro lado, la integración del IPD y su entrega se realizaron

## 10.4. Seguimiento de las horas trabajo

puntualmente. Estos incrementos en las horas de trabajo no tuvieron un impacto negativo en el diagrama de Gantt ni en la fecha de entrega prevista, ya que se mantuvieron dentro del plazo establecido.

### 10.4.1.3. Ultimo análisis de las horas de trabajo

En este último análisis de las horas de trabajo se llevó a cabo después de la finalización y cierre del proyecto. Este análisis nos proporciona información general sobre cómo se llevaron a cabo las tareas en su fase final, brindándonos una visión retrospectiva. Esto nos ofrece valiosas lecciones que podemos aplicar en proyectos futuros para mejorar nuestra planificación y ejecución en proyectos de tamaño similar.

Paquete de Trabajo	Tarea	Estimación en horas (h)	Real en horas (h)
<b>Paquete de trabajo Análisis y Planificación (AP)</b>	AP.1 Análisis del Dominio (AD)	4	2
	AP.2 Planificación (P)	25	25
	AP.3 Análisis de las Herramientas (AH)	5	3
	<b>SUBTOTAL AP</b>	<b>34</b>	<b>30</b>
<b>Paquete de trabajo Desarrollo Incremento 1</b>	I1.1 Análisis de Objetivos y Requisitos (AOR)	5	5
	I1.2 Expansión de los Requisitos y Objetivos (ERO)	5	3
	I1.3 Creación de Diagramas (CD)	20	20
	I1.4 Diseño de la Solución y Arquitectura del sistema	10	5
	<b>SUBTOTAL I1</b>	<b>40</b>	<b>33</b>
<b>Paquete de trabajo Desarrollo Incremento 2-Base de Datos (BD)</b>	BD.1 Análisis de la Esquema de la Base de Datos (AE)	5	4
	BD.2 Diseño de la Base de Datos (DBD)	15	15
	BD.3 Implementación de la Base de Datos (IBD)	10	8
	BD.4 Pruebas sobre la Base de Datos (PBD)	10	8
	BD.5 Creación del Entorno de Producción (CEP)	6	3
	BD.6 Lanzamiento de la Base de Datos Inicial (LBD)	5	3
	<b>SUBTOTAL BD</b>	<b>51</b>	<b>41</b>
<b>Paquete de trabajo Desarrollo Incremento 3-Interfaz de Procesamiento de Datos (IPD)</b>	IPD.1 Incremento necesario de la BD	5	3
	IPD.2 Implementación de la interfaz.	30	31
	IPD.3 Pruebas	15	26
	IPD.4 Integración y entrega del IPD.	10	10
	<b>SUBTOTAL IPD</b>	<b>60</b>	<b>70</b>
<b>Paquete de trabajo Desarrollo Incremento 4-Software de Visualización de Datos (SVD)</b>	SVD.1 Análisis de los requisitos para el SVD.	5	2
	SVD.2 Incremento necesario de la BD	5	2
	SVD.3 Implementación del SVD	20	20
	SVD.4 Diseño de la aplicación	20	20
	SVD.5 Pruebas.	10	10
	SVD.6 Integración y entrega del SVD.	2	2
	<b>SUBTOTAL SVD</b>	<b>62</b>	<b>56</b>
<b>Paquete de trabajo Desarrollo Incremento 5-Vision por Computadora</b>	VC.1 Análisis de algoritmos y herramientas.	3	3
	VC.2 Implementación del VC	8	5
	VC.3 Pruebas.	2	2
	VC.4 Integración y entrega del VC.	2	1.5
	<b>SUBTOTAL VC</b>	<b>15</b>	<b>11.5</b>
<b>Paquete de trabajo Pruebas Sistema Completo</b>	<b>SUBTOTAL PSC</b>	<b>20</b>	<b>20</b>
<b>Paquete de trabajo Gestión de Proyecto (GP)</b>	GP.1 Seguimiento y Control (SyC)	10	10
	GP.2 Reuniones.	8	9
	<b>SUBTOTAL GP</b>	<b>18</b>	<b>19</b>
<b>Paquete de trabajo Memoria</b>	MEM.1 Recopilación de información y datos relevantes	80	85
	MEM.2 Preparación para la defensa final del proyecto	7	3
	<b>SUBTOTAL MEM</b>	<b>87</b>	<b>88</b>
<b>HORAS TOTALES</b>		<b>387</b>	<b>368.5</b>
<b>LIMITE HORAS TOTALES</b>		<b>360</b>	
		<b>SOBRESALE</b>	<b>DENTRO DEL LIMITE</b>
		27	0

**Figura 10.7:** Ultimo análisis de horas

En total, se llevaron a cabo 9 reuniones en lugar de las 8 originalmente planificadas debido al incremento 5. En cuanto al trabajo de la memoria, se requirieron 5 horas adicionales

## 10. SEGUIMIENTO Y CONTROL

---

debido a correcciones. En general, se respetó el plan de trabajo establecido, superando el límite previsto del trabajo de fin de grado solo por 8 horas.

### 10.5. Riesgos y mitigación

#### 10.5.1. R-5: Riesgo de Empleabilidad

Este riesgo se mencionó debido a las discusiones durante la cuarta reunión (ver 12.4), donde se observó que el alumno estaba centrando más su atención en el trabajo de fin de grado, lo que requería una evaluación de posibles cambios significativos.

- **Descripción:** Este riesgo se refiere a la posibilidad de que el alumno encuentre empleo antes de completar el proyecto.
- **Plan de Prevención:** No se aplicó ningún plan de prevención específico.
- **Impacto:** Bajo (Impacto insignificante).
- **Cambios en la Planificación:** No se realizaron cambios en la planificación debido a este riesgo.

#### 10.5.2. R-3: Riesgo de Tiempo

Este riesgo se mencionó a raíz de las conversaciones sostenidas durante la sexta reunión (consultar 12.4), en la que se destacó que nos encontramos en la fase final del proyecto y solo quedan tres semanas para cumplir con los plazos establecidos.

- **Descripción:** Posibilidad de no cumplir con el plazo establecido.
- **Plan de Prevención:** Creación de un cronograma realista y seguimiento regular del progreso.
- **Impacto:** Medio
- **Cambios en la Planificación:** Se realizó una replanificación 2 (ver 10.3.2) que nos permitió gestionar mejor el tiempo y se programaron las dos últimas reuniones para un plazo más breve.

## CAPÍTULO

# 11 |

## Comparación en el entorno de sistemas de alto nivel

En este capítulo, se presenta un resumen detallado del sistema desarrollado, siguiendo las mejores prácticas a nivel global. También, se ofrecerá una visión de los desafíos inherentes y se explorará la estructura del sistema en relación con entornos empresariales a nivel mundial.

### 11.1. Resumen del Sistema

Dando un paso significativo, en esta sección examinamos minuciosamente el sistema en su totalidad, otorgando relevancia a cada una de sus partes. Además, comparamos este sistema con otros sistemas de tamaño similar que existen en la realidad. Aunque el sistema actual representa el punto de partida de lo que es posible, marca el inicio de futuras exploraciones y desarrollos del sistema.

Durante el desarrollo del sistema, hemos aprovechado tecnologías de vanguardia para explorar la creación de sistemas altamente escalables. Desde el backend, hemos empleado el framework Spring, mientras que en el frontend hemos evaluado la versatilidad de Thymeleaf y JavaScript. Además de estas tecnologías, hemos implementado nuestro propio pipeline de Integración Continua/Entrega Continua (CI/CD) en Docker, haciendo uso de la última tecnología DevOps con Jenkins y Maven.

En cuanto a las pruebas y la calidad del código, hemos utilizado herramientas esenciales como JUnit 5 y Mockito para las pruebas unitarias, así como Postman para la verificación de servicios. Estas prácticas son fundamentales en el desarrollo de sistemas robustos y fiables.

Además de la variedad tecnológica, hemos integrado diversas APIs, como Plate Recognizer y Stripe API, para complementar el sistema y prepararlo para futuros despliegues en entornos empresariales. Además de la elección de tecnologías, hemos aplicado una optimización de código eficiente, haciendo uso de estructuras de datos adecuadas, como los mapas para el algoritmo de reconocimiento, y patrones de diseño como el Facade, respetando el principio de Responsabilidad Única (SRP).

Estos esfuerzos combinados resultan en la creación de un sistema realista y potencialmente investigable, con la capacidad de desplegarse a nivel nacional, lo que lo convierte en un proyecto poderoso. A continuación, se detallará cada parte del proyecto, incluida su evolución durante el desarrollo.

### **11.1.1. CI/CD pipeline**

La implementación de nuestro propio pipeline de desarrollo nos ha brindado una perspectiva renovada sobre la construcción de sistemas y proyectos. La potente capacidad de automatizar la integración continua y el desarrollo nos permite identificar y abordar errores de manera más ágil, antes de que lleguen al entorno de producción. Aunque no hemos alcanzado por completo nuestro objetivo principal con el pipeline, debido a restricciones tecnológicas en el entorno local, hemos abierto una puerta hacia el futuro desarrollo del proyecto, al habilitar la automatización del código.

Nuestro pipeline, aunque sencillo, representa un punto de partida que explora las infinitas posibilidades de investigación y aplicación de estos conceptos.

### **11.1.2. Servicios RESTful**

En la actualidad, prácticamente todas las empresas brindan acceso a sus API, y la exploración e investigación en este ámbito, que permite a otros usuarios aprovechar nuestro servicio, abre la puerta a una perspectiva invaluable sobre cómo los datos pueden ser utilizados de manera creativa. Este servicio REST, si se considera en el futuro, puede incluir límites de uso, similar a muchas otras plataformas, como X (anteriormente Twitter), OpenAI, Reddit, Amazon, e incluso Plate Recognizer, que es la API que utilizamos y que permite hasta 2500 llamadas al mes. Todo esto nos ayuda a comprender el desarrollo de aplicaciones completas y las posibilidades que se abren.

Además, en nuestro servicio, ofrecemos transparencia al usuario al permitirle visualizar las secciones posibles y los pasos a través de dichas secciones, lo que enriquece aún más la experiencia del usuario manteniendo la seguridad.

### **11.1.3. IPD simulador**

Lamentablemente, no podemos desplegar ni simular el sistema en un entorno real por varias razones. En primer lugar, la limitación de tiempo ha centrado nuestro proyecto en la investigación de sistemas empresariales, específicamente un prototipo de sistema de peaje. Este prototipo representa un punto de partida, una semilla que ilustra las posibilidades. El trabajo realizado en tres meses abarca desde la fase de planificación hasta la entrega, un proceso que requiere una inversión de tiempo significativa. El desarrollo de un sistema de esta magnitud en solitario es prácticamente imposible, ya que abarca diversas etapas y un total de 360 horas de trabajo.

La segunda razón es la escasez de recursos y personal. Proyectos de esta envergadura demandan un equipo multidisciplinario que incluya expertos en ingeniería de software, inteligencia artificial, hardware y computación, lo que exige un conjunto excepcional de conocimientos y habilidades. La decisión de desarrollar y presentar un simulador se basa en la necesidad de poder simular el comportamiento del sistema. En la realidad, muchas empresas utilizan sistemas de demostración para mostrar a sus clientes cómo funcionan sus

soluciones. En nuestro caso, considerando el escenario hipotético, estos clientes podrían ser la DGT y el gobierno español. El simulador ofrece una representación visual de cómo se almacenarán los datos y proporciona información suficiente para validar el sistema.

#### **11.1.4. SVD Web Application**

Dentro del contexto del sistema, la idea central consistía en proporcionar a los usuarios la capacidad de visualizar sus viajes, conocer los costos asociados, y acceder a rutas y estadísticas esenciales. La aplicación web cumple exitosamente con este propósito, logrando así la conexión entre el sistema de peaje y una interfaz web amigable para el usuario.

Este aspecto no solo enriquece la experiencia del usuario, sino que también destaca la importancia del diseño de interfaz de usuario y experiencia de usuario (UI/UX). Se ha dedicado un esfuerzo significativo a esta área para satisfacer las expectativas visuales y funcionales de los usuarios, asegurando así que la aplicación sea atractiva y fácil de usar.

#### **11.1.5. Consideraciones de Seguridad y Privacidad**

Todas las aplicaciones deben priorizar la seguridad y la privacidad de los usuarios, considerando estos aspectos como fundamentales. En mi opinión, son uno de los tres factores más importantes tanto durante el desarrollo como en el despliegue del sistema. En nuestro sistema, hemos implementado medidas de seguridad en tres áreas clave: el uso del simulador, la aplicación web y los servicios.

En cuanto a la seguridad, hemos empleado una sólida función hash, BCrypt, ampliamente reconocida y utilizada en la actualidad. Además, para garantizar la seguridad en la autenticación y autorización de los servicios, hemos implementado la generación de tokens JWT (JSON Web Tokens). Estas medidas contribuyen a fortalecer la seguridad del sistema y a cerrar posibles brechas de seguridad.

La privacidad también es un aspecto crucial que respetamos. En los servicios REST, hemos diseñado la obtención de datos de usuarios de manera que, como se menciona anteriormente, los datos se anonimizan mediante el enmascaramiento de las matrículas de vehículos. De esta manera, aseguramos que la información personal de los usuarios esté protegida y que se respete su privacidad en todo momento.



# 12

## CAPÍTULO

# Conclusiones

En este último capítulo, se reflexiona sobre los logros, el cumplimiento de los objetivos y el trabajo futuro, así como la visión a largo plazo del proyecto y la conclusión de lo realizado.

### 12.1. Logros

Hemos logrado desarrollar un sistema funcional que simula el flujo de un sistema de peaje, proporcionando al usuario información sobre su ubicación y el costo de los datos. Además, hemos mejorado la representación de las autopistas en comparación con el sistema web de ViaT.

### 12.2. Cumplimiento de los Objetivos

En esta sección, se evalúa el grado de cumplimiento de los objetivos, tanto los principales como los secundarios, y de los requisitos del proyecto.

#### 12.2.1. Objetivos Principales

1. Se ha logrado satisfactoriamente el desarrollo de una interfaz de reconocimiento de matrículas.
2. La creación de una aplicación web segura que permite el registro y el reconocimiento de usuarios ha sido exitosa.

#### 12.2.2. Objetivos Secundarios

1. Se ha alcanzado con éxito la implementación de un algoritmo propio de Reconocimiento de Matrículas (LPR, por sus siglas en inglés).
2. El desarrollo de software que puede almacenar la ubicación, la hora y el tiempo de permanencia en la autopista ha sido satisfactorio.

## 12. CONCLUSIONES

---

3. El software es capaz de reconocer las etiquetas de emisiones de CO<sub>2</sub> de los vehículos nacionales, cumpliendo con este objetivo secundario.

### 12.3. Trabajo Futuro

#### 12.3.1. Arquitectura de Microservicios

Para futuras iteraciones, se considera la implementación de una arquitectura de microservicios. Esto permitirá una mayor escalabilidad y modularidad del sistema, lo que facilitará la incorporación de nuevas funcionalidades y la gestión de componentes específicos de manera independiente.

#### 12.3.2. Base de Datos Distribuidas y Balanceo de Carga

Se plantea la posibilidad de migrar hacia una base de datos distribuida para manejar el crecimiento esperado de datos. Además, se explorará el uso de técnicas de balanceo de carga para garantizar un rendimiento óptimo y una alta disponibilidad del sistema.

#### 12.3.3. Mejorando el Frontend: React

Para mejorar la experiencia del usuario, se considera la adopción del framework React para el desarrollo del frontend. Esto permitirá una interfaz más dinámica y receptiva, así como la incorporación de características modernas de diseño.

#### 12.3.4. CI/CD Pipeline - Parte 2

Se planea expandir el proceso de Integración Continua y Entrega Continua (CI/CD) para incluir pruebas más exhaustivas y automatizar aún más el despliegue. Esto garantizará la estabilidad y la entrega eficiente de nuevas versiones del sistema.

#### 12.3.5. Despliegue del Sistema

En el futuro, se podría llevar a cabo la implementación y despliegue del sistema en un entorno de producción. Esto incluirá la configuración de servidores, la gestión de recursos y la monitorización para asegurar un funcionamiento sin problemas y una alta disponibilidad.

### 12.4. Conclusiones

Finalmente, el desarrollo de un sistema de alto nivel como este abre nuevas oportunidades que abarcan desde la planificación, vista en la gestión de proyectos y fundamental en cualquier proyecto de cualquier tamaño, hasta la seguridad en la gestión de sistemas y privacidad de los datos, así como la creación de algoritmos de visión por computadora. Este proyecto abarca diversas especialidades y las relaciona de manera sólida en el ámbito de la ingeniería de software. Se ha analizado e investigado el desarrollo y la arquitectura posibles de un sistema de telepeaje automático, además de implementar las mejores prácticas y tomar decisiones acertadas sobre el uso del sistema. También se exploró el uso de la automatización de procesos y la conexión con diferentes APIs, así como la mejora en el flujo y la fluidez de los datos. Por último, se ha investigado la posibilidad de crear nuestro

#### 12.4. Conclusiones

propio algoritmo y se ha preparado para el futuro despliegue del sistema. En resumen, se ha desarrollado un proyecto desde cero, siguiendo las mejores prácticas y enfocándose en diferentes aspectos clave.



# **Apéndice**



# **Actas de Reuniones**

## **Anexo: Actas de Reuniones del Proyecto**

---

Este anexo recopila todas las actas de las reuniones que se han realizado durante el proyecto. El objetivo de estas actas es agrupar los puntos más importantes de las reuniones. A raíz de ellas, se puede tener una trazabilidad de todo el proyecto. Estas actas se han realizado con los interesados del proyecto y son redactadas por el alumno.

Todas las actas siguen una plantilla, donde se recopila en primer lugar información básica (p. ej.: hora de inicio, asistentes...) y luego se separan los puntos más importantes en diferentes apartados. Los puntos más importantes son:

- Orden del día: información anterior a la reunión, indicando el objetivo de la misma.
- Resumen de la reunión: se resume todas las fases de la reunión, sin indicar las decisiones tomadas.
- Estado del proyecto: sitúa el proyecto dentro de uno de los paquetes que se ha definido en la planificación.
- Decisiones: decisiones que se han tomado para el futuro del proyecto, y se enumeran las tareas que tienen que realizar los interesados de la reunión.

## Plantilla de Actas de Reuniones del Proyecto

**KA-TFG-AR-tt-nn-dd/mm/yyyy**

Donde:

- tt refleja el título de la reunión
- nn indica el número de la reunión comenzando desde 00

**Fecha:** [Fecha de la reunión]

**Hora de inicio:** [Hora de inicio]

**Duración:** [Duración de la reunión]

**Asistentes:** [Lista de asistentes]

**Tipo de Reunión:** [Tipo de Reunión]

## Orden del día

[Detalles del orden del día de la reunión]

## Resumen de la reunión

[Resumen de las fases de la reunión, sin incluir las decisiones tomadas]

## Estado del proyecto

[Situación actual del proyecto y su relación con los paquetes definidos en la planificación]

## Decisiones

Decisión 1 [Tareas asignadas a los interesados de la reunión relacionadas con la decisión 1]

Decisión 2 [Tareas asignadas a los interesados de la reunión relacionadas con la decisión 2]

Próxima Reunión [Decisión de la siguiente reunión]

## **KA-TFG-AR-Inicion-Proyecto-00-14/07/2023**

**Fecha:** 14/07/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Maider Azanza y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del día**

El orden del día es poner en marcha el proyecto y resolver dudas.

### **Resumen de la reunión**

En la reunión se han resuelto dudas sobre, la planificación, formatos, detalles de los conjuntos de datos y el SPD y repaso a los tutores sobre el proyecto.

### **Estado del proyecto**

El proyecto esta en una fase inicial.

### **Decisiones**

- Formatos de los documentos: Se ha decidido utilizar el formato Látex para la planificación y la mayoría de los documentos.
- Riesgo de encontrar trabajo en Septiembre: Se ha establecido que hay un riesgo de encontrar trabajo en Septiembre por parte del alumno.
- Conjuntos de datos: Se han analizado diferentes conjuntos de datos para facilitar el inicio del proyecto del primer incremento del SPD.
  - [UCSD](#)
  - [OpenALRP](#)
  - [ALPR, ANPR](#)
  - [UFPR-ALPR](#)
  - [CCPD](#)
  - [Plate Recognizer](#)
- Próxima Reunión: Se ha decidido que la siguiente reunión sera el 27/07/2023 o en una fecha antes para hacer las correcciones de la planificación.

## **KA-TFG-AR-Seguimiento-y-Control-01-27/07/2023**

**Fecha:** 27/07/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Maider Azanza y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del día**

Seguimiento y control, correcciones sobre la planificación inicial y dudas sobre uso de api de terceros.

### **Resumen de la reunión**

En la reunión se han resuelto preguntas sobre el uso de api de terceros y dudas sobre la corrección de la planificación.

### **Estado del proyecto**

El proyecto esta en una fase inicial.

### **Decisiones**

- Uso de api de terceros: Se ha decidido utilizar una api de terceros para el reconocimiento de matriculas en una fase inicial y en una fase posterior crear nuestro propio algoritmo si se considera oportuno.
- Preguntas sobre la corrección de la planificación inicial: Se han resuelto diferentes preguntas sobre la planificación inicial, como, apartados que se deben de modificar, eliminar y añadir.
- Próxima Reunión: La próxima reunión se ha decidido para el 05/09/2023, con fecha y horario tentativos.

## **KA-TFG-AR-Seguimiento-y-Control-02-05/09/2023**

**Fecha:** 05/09/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Maider Azanza y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del día**

Seguimiento y control, segunda corrección sobre la planificación y análisis sobre los cambios y lo realizado hasta este momento.

### **Resumen de la reunión**

Se ha realizado una prueba de la API seleccionada para el reconocimiento de matrículas, confirmando su correcto funcionamiento. Durante el proceso, se ha identificado un retraso en el proyecto que debe reflejarse en el seguimiento y control del mismo. Además, se ha tomado la decisión de postergar la implementación de la parte de visión por computadoras hasta una etapa posterior, específicamente en el Incremento 5.

### **Estado del proyecto**

El proyecto debería de estar en el incremento 2, pero aun esta en incremento 1. Retraso de 3 tareas importantes.

### **Decisiones**

- Registrar los retrasos detectados en la memoria de seguimiento y control, incluyendo los cambios identificados y sus respectivos impactos.
- Proseguir con la finalización del Incremento 1 y dar inicio al Incremento 2.
- Planificar el Incremento 5: Este incremento se centrará en el desarrollo de un algoritmo personalizado y la captura de las etiquetas ambientales de forma más detallada (fase optativa).
- Próxima Reunión: La próxima reunión se ha decidido realizar el día 19/09/2023 a las 12:00 EET.

## KA-TFG-AR-Seguimiento-y-Control-03-19/09/2023

**Fecha:** 19/09/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Maider Azanza y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### Orden del día

Seguimiento y control, información sobre el retraso identificado y incremento 1 finalizado.

### Resumen de la reunión

Se ha confirmado la finalización del primer incremento y la continuación con el segundo incremento. Se espera una confirmación formal en la próxima reunión con respecto a la captura de los requisitos iniciales. Además, se ha aclarado que el proyecto sigue estando dentro del plazo previsto, pero se considera necesario replanificarlo para asegurar la presentación del proyecto en noviembre. Esto implica adelantar el tercer incremento. También se ha identificado que las horas de trabajo no se están siguiendo estrictamente, lo que ha llevado a realizar un análisis detallado de las horas trabajadas hasta el momento.

### Estado del proyecto

El proyecto continúa dentro del plazo inicialmente planificado, sin desviaciones en este momento. Actualmente, nos encontramos en el Incremento 2 de la planificación del proyecto.

### Decisiones

- Registrar con precisión las horas de trabajo en lugar de estimarlas, de modo que se pueda llevar un seguimiento concreto y preciso.
- Anticipar la ejecución del Incremento 3 y realizar una reprogramación del diagrama Gantt y las horas de trabajo.
- Inicialmente, se había asumido que el estudiante podría encontrar trabajo, pero se ha modificado el plan y ahora el estudiante ha dado prioridad al Trabajo Fin de Grado (TFG). Es necesario evaluar si esto tiene impactos significativos en el proyecto.
- Esperar la validación sobre la captura de requisitos.
- El Incremento 2 continúa en curso, ya que no es necesario esperar hasta la próxima reunión debido al uso de la tecnología Hibernate, que facilita las modificaciones.

- Próxima Reunión: La próxima reunión se ha decidido realizar el día 25/09/2023 a las 11:00 EET.

## **KA-TFG-AR-Seguimiento-y-Control-04-25/09/2023**

**Fecha:** 25/09/2023

**Hora de inicio:** 10:00 CET, 11:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Maider Azanza y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del día**

Seguimiento y control, revisión sobre el capítulo de captura de requisitos, concretamente las diagramas.

### **Resumen de la reunión**

Se han comentado los problemas identificados en el capítulo de captura de requisitos y las diagramas de casos de uso y de dominio.

### **Estado del proyecto**

El proyecto sigue en linea con el incremento 2.

### **Decisiones**

- Se ha tomado la decisión de identificar de manera concreta los actores del sistema actual y clarificar los stakeholders en el capítulo de visión general. Además, se ha decidido hacer una referencia a la imagen dentro de este capítulo.
- Se ha completado el diagrama de casos de uso y el capítulo de captura de requisitos con los siguientes cambios:
  - Se ha optado por extraer los requisitos funcionales y no funcionales de la planificación y agregarlos al apartado de captura de requisitos. De esta manera, se evita la separación entre la planificación y la captura de requisitos.
  - Se ha definido con precisión la lista de actores, eliminando aquellos que no son necesarios.
  - Se ha decidido proporcionar una explicación detallada para cada caso de uso, incluyendo un identificador para facilitar su identificación.
- Se ha acordado incluir un capítulo de trabajo futuro en el que se puedan documentar algunas de las decisiones relacionadas con los casos de uso y el dominio.
- Es necesario realizar correcciones en el diagrama de la base de datos.
- Se continúa con el incremento 2 de la base de datos, detallando los modelos del proyecto y creando el inicializador de la base de datos para realizar pruebas de datos.

- Se ha optado por hacer un supuesto en relación al sistema, específicamente en lo que respecta a los datos de tramos, puntos, y otros elementos.
- Próxima Reunión: Se concretara el día 29/09/2023 (viernes) por correo electrónico.

## **KA-TFG-AR-Seguimiento-y-Control-05-10/10/2023**

**Fecha:** 10/10/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Maider Azanza y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del día**

Seguimiento y control, cierre del incremento 3 y demostración de la funcionalidad.

### **Resumen de la reunión**

Se ha revisado lo que se ha realizado desde el ultimo seguimiento y control. Se ha realizado una demostración "en vivo" para cerrar el incremento 3 y poder continuar con el ultimo incremento.

### **Estado del proyecto**

El proyecto entra en la ultima etapa: incremento 4

### **Decisiones**

- Se ha tomado la decisión de especificar con mayor detalle el concepto del módulo IPD. En particular, se basa en la creación de un simulador para el sistema de peaje.
- Se ha identificado un riesgo de alto nivel: solo quedan 3 semanas para completar el proyecto.
- Los próximos pasos incluyen el Incremento 4, la revisión de los capítulos uno referentes a Pruebas y Métricas, así como la inclusión de los servicios REST desarrollados.
- Próxima Reunión: La próxima reunión se ha decidido realizar el día 16/10/2023 (Lunes) a las 12:00 EET.

## **KA-TFG-AR-Seguimiento-y-Control-06-16/10/2023**

**Fecha:** 16/10/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Maider Azanza y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del día**

Seguimiento y control general.

### **Resumen de la reunión**

Se ha revisado lo que se realizó durante el periodo de la ultima reunión.

### **Estado del proyecto**

El proyecto sigue en el ultimo incremento 4 siguiendo el plan.

### **Decisiones**

- Se ha decidido corregir algunos capítulos de la memoria y continuar con el desarrollo.
- Próxima Reunión: Se decidió realizar dos reuniones una relacionada con el incremento 5 del VC que se realizara el día 19/10/23 (Jueves) a las 12 EET y la otra reunión realizar el día 23/10/2023 (Lunes) a las 12:00 EET.

## **KA-TFG-AR-LPR-07-19/10/2023**

**Fecha:** 19/10/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev y Ekaitz Jauregi

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del día**

Trabajo sobre el algoritmo del LPR (License Plate Recognition)

### **Resumen de la reunión**

Se ha revisado lo que se realizó durante el periodo de la ultima reunión.

### **Estado del proyecto**

El proyecto sigue en el ultimo incremento 4.

### **Decisiones**

- No había

## **KA-TFG-AR-Cierre-de-proyecto-08-23/10/2023**

**Fecha:** 23/10/2023

**Hora de inicio:** 11:00 CET, 12:00 EET

**Duración:** 1h

**Asistentes:** Konstantin Andreev, Ekaitz Jauregi y Maider Azanza

**Tipo de Reunión:** Online: Microsoft Teams

### **Orden del Día**

En esta reunión, se abordarán los siguientes temas:

- Conclusión del proyecto y revisión del trabajo realizado.

### **Resumen de la Reunión**

Durante la reunión, se llevó a cabo una demostración completa del sistema y se llegó a la conclusión de que el proyecto se encuentra en su fase final.

### **Estado del Proyecto**

El proyecto ha alcanzado su etapa de cierre y finalización.

### **Decisiones y Tareas**

En la reunión no se tomaron nuevas decisiones ni se asignaron tareas adicionales.



# Bibliografía

- [1] [https://media.wired.com/photos/5a14829a63f8d15055c8070b/master/w\\_1920,c\\_limit/toll-TA.jpg](https://media.wired.com/photos/5a14829a63f8d15055c8070b/master/w_1920,c_limit/toll-TA.jpg). Ver páginas x, 1.
- [2] <https://www.boe.es/doue/2016/119/L00001-00088.pdf>. Ver página 7.
- [3] <https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673>. Ver página 7.
- [4] <https://creativecommons.org/licenses/by-nc-nd/3.0/es/>. Ver página 7.
- [5] <https://www.overleaf.com/>. Ver página 20.
- [6] Michael J. Hernandez. *Database Design for Mere Mortals*. Addison-Wesley Professional, Boston, MA, 4th edition, 2013. Ver página 34.
- [7] <https://www.decipherzone.com/blog-detail/benefits-web-application-development>. Ver página 38.
- [8] <https://www.decipherzone.com/blog-detail/security-web-app-development>. Ver página 39.
- [9] <https://getbootstrap.com/>. Ver página 41.
- [10] <https://react.dev/>. Ver página 41.
- [11] <https://angular.io/>. Ver página 41.
- [12] Craig Walls. *Spring in Action*. Manning Publications, Shelter Island, NY, 2019. Ver página 44.
- [13] <https://www.mysql.com/>. Ver página 44.
- [14] <https://www.python.org/>. Ver página 45.
- [15] <https://projectlombok.org/>. Ver página 47.
- [16] <https://www.docker.com/>. Ver página 48.
- [17] <https://matriculasdelmundo.com/europeas.php>. Ver página 80.