

# Memoria del Proyecto

## PLATAFORMA PARA EL ANÁLISIS DE LA DENSIDAD DE CARROS DE LA COMPRA EN UN SUPERMERCADO

Trabajo de Fin de Grado  
INGENIERÍA INFORMÁTICA



**VNiVERSiDAD  
D SALAMANCA**

Julio 2023

AUTOR

Javier Caballero Sandoval

TUTORES

Héctor Sánchez San Blas

Gabriel Villarrubia González

# Certificado de los Tutores

D. Héctor Sánchez San Blas y D. Gabriel Villarrubia González, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo titulado "Plataforma para el análisis de la densidad de carros de la compra en un supermercado" ha sido realizado por D. Javier Caballero Sandoval, con el número de documento \*\*\*\*7295S y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo Fin de Grado de la Titulación Grado de Ingeniería Informática en esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 28 de junio de 2023

D. Héctor Sánchez San Blas

D. Gabriel Villarrubia Sánchez



## Resumen

En la actualidad, los supermercados ofrecen una gran oferta de productos y disponibilidad de horarios. Esto provoca, como consecuencia, el aumento de personas que realizan la compra asiduamente en un supermercado. Por ello, la organización y control de un supermercado es más complejo en la actualidad.

Para mejorar diversos factores como la eficiencia, eficacia, experiencia de los clientes o seguridad es necesario recoger información de diferentes aspectos relativos al flujo de las clientes a lo largo del supermercado, entre ellos los carros de la compra.

El principal objetivo de este proyecto es el diseño y desarrollo de un sistema de detección y trackeo de los carros de la compra a través de diferentes zonas de un supermercado sin la necesidad de incluir un dispositivo en cada uno de los carros. Además, se desarrolla una aplicación web que permite observar de manera gráfica los datos recogidos acerca de los carros permitiendo su posterior tratamiento para mejorar los diferentes aspectos del supermercado, así como la visualización en tiempo real de las cámaras instaladas para tener la capacidad de observar si se ha producido una incidencia de manera rápida.

Por otra parte, la aplicación permite la personalización para cada uno de los supermercados que adopte el sistema ya que posee la capacidad de personalizar las zonas del supermercado, las cámaras instaladas y los avisos deseados de detectar.

**Palabras claves:** detección de objetos, carro de la compra, trackeo, YOLO, compra inteligente.



## Summary

Nowadays, supermarkets offer a wide range of products and opening hours. As a result, more and more people shop regularly in supermarkets. The organisation and control of a supermarket is therefore more complex nowadays.

In order to improve various factors such as efficiency, effectiveness, customer experience or safety, it is necessary to collect information on different aspects of customer flow throughout the supermarket, including shopping trolleys.

The main objective of this project is the design and development of a system for detecting and tracking shopping trolleys through different areas of a supermarket without the need to include a device in each of the trolleys. In addition, a web application is developed that allows the data collected about the trolleys to be observed graphically, allowing its subsequent processing to improve the different aspects of the supermarket, as well as the real-time visualisation of the cameras installed to be able to observe if an incident has occurred quickly.

Moreover, the application allows customisation for each of the supermarkets that adopt the system as it has the ability to customise the areas of the supermarket, the cameras installed and the desired warnings to be detected.

**Keywords:** object detection, shopping trolley, tracking, YOLO, smart shopping.

# Tabla de Contenido

1. <b>Introducción</b> .....	1
2. <b>Estudio del Estado del Arte</b> .....	3
2.1. Investigación.....	3
2.2. Soluciones Comerciales.....	3
3. <b>Objetivos</b> .....	6
3.1. Objetivos del Sistema.....	6
3.2. Objetivos Personales.....	6
4. <b>Conceptos Teóricos</b> .....	8
4.1. Detección de Objetos.....	8
4.2. Trackeo de Objetos.....	8
4.3. Rest.....	8
5. <b>Técnicas y Herramientas</b> .....	9
5.1. Tracker.....	9
5.1.1. YOLO v5.....	9
5.1.2. Roboflow.....	9
5.1.3. Google Colab.....	10
5.1.4. Python.....	10
5.1.5. Flask.....	10
5.1.6. OpenCV.....	10
5.1.7. NumPy.....	10
5.1.8. Dlib.....	11
5.2. Aplicación Web.....	11
5.2.1. JavaScript.....	11
5.2.2. HTML.....	11
5.2.3. CSS.....	11
5.2.4. Vue.js.....	11
5.2.5. Vuetify.....	12
5.2.6. PrimeVue.....	12
5.2.7. Axios.....	12
5.2.8. Vue-chartjs.....	12
5.3. Servidor.....	12
5.3.1. NodeJS.....	12
5.3.2. NPM.....	13
5.3.3. Express.....	13
5.3.4. MySQL.....	13
5.3.5. Sequelize.....	13
5.4. Editor de Texto.....	13
5.4.1. Visual Studio Code.....	13

5.5.	Herramientas Case.....	14
5.5.1.	Microsoft Project.....	14
5.5.2.	Visual Paradigm .....	14
5.5.3.	EZestimate .....	14
5.6.	Herramientas de Generación de Documentación .....	14
5.6.1.	Sphinx.....	14
5.6.2.	JSDoc .....	14
5.6.3.	Vue Styleguidist.....	14
<b>6.</b>	<b>Aspectos Relevantes del Desarrollo .....</b>	<b>15</b>
6.1.	Marco de Trabajo .....	15
6.2.	Estimación de la Duración del Proyecto.....	16
6.3.	Planificación Temporal .....	18
6.4.	Especificación de Requisitos.....	19
6.4.1.	Participantes.....	19
6.4.2.	Objetivos del Sistema.....	19
6.4.3.	Requisitos de Información .....	20
6.4.4.	Requisitos Funcionales .....	21
6.4.5.	Requisitos No Funcionales.....	23
6.5.	Análisis de Requisitos .....	24
6.5.1.	Modelo de Dominio.....	24
6.5.2.	Realización de Casos de Uso Análisis.....	25
6.5.3.	Clases de Análisis y Propuesta de Arquitectura .....	25
6.6.	Diseño del Sistema.....	26
6.6.1.	Patrones Arquitectónicos y de Diseño .....	26
6.6.2.	Patrón Modelo -Vista- Modelo de Vista (MVVM) .....	26
6.6.3.	Patrón DAO .....	27
6.6.4.	Subsistemas de Diseño.....	27
6.6.5.	Clases de Diseño .....	28
6.6.6.	Realización de Casos de Uso Diseño .....	32
6.6.7.	Diseño de la Base de Datos .....	34
6.6.8.	Modelo de Despliegue.....	34
6.7.	Implementación .....	35
6.8.	Pruebas .....	35
6.9.	Funcionalidad del Sistema .....	36
6.9.1.	Usuario No Logueado.....	36
6.9.2.	Usuario .....	39
6.9.3.	Administrador .....	44
<b>7.</b>	<b>Resultados Entrenamiento Modelo .....</b>	<b>53</b>
<b>8.</b>	<b>Limitaciones del Sistema .....</b>	<b>55</b>
<b>9.</b>	<b>Conclusiones y Líneas de Trabajo Futuras .....</b>	<b>56</b>

9.1. Conclusiones .....	56
9.2. Líneas de Trabajo Futuras .....	56
10. <b>Bibliografía</b> .....	57

## Índice de Figuras

Figura 1: Distrito K.....	4
Figura 2: Wyn Enterprise.....	5
Figura 3: Roboflow .....	9
Figura 4: Diagrama de Proceso Unificado .....	16
Figura 5: Estimación del Esfuerzo.....	16
Figura 6: Factores de Complejidad Técnica .....	17
Figura 7: Factores de Entorno.....	17
Figura 8: Planificación Temporal.....	18
Figura 9: Diagrama de Paquetes .....	21
Figura 10: Actores del Sistema .....	21
Figura 11: Diagrama de Casos de Uso Autenticación de Usuarios.....	22
Figura 12: Modelo del Dominio .....	24
Figura 13: Diagrama de Secuencia Análisis Registrar Usuario .....	25
Figura 14: Propuesta de Arquitectura .....	25
Figura 15: Modelo-Vista-ModeloVista .....	26
Figura 16: DAO .....	27
Figura 17: Subsistemas de Diseño .....	28
Figura 18: ControllerApi .....	29
Figura 19: DAOApi .....	29
Figura 20: Model.....	30
Figura 21: Components.....	30
Figura 22: Views.....	31
Figura 23: Tracker .....	31
Figura 24: ServerCollector .....	32
Figura 25: Diagrama de Secuencia Diseño Registrar Usuario .....	33
Figura 26: Diseño de la Base de Datos .....	34
Figura 27: Modelo de Despliegue .....	34
Figura 28: Pantalla Iniciar Sesión.....	37
Figura 29: Pantalla Registrar Usuario .....	37
Figura 30: Pantalla Correo Electrónico Recuperar Contraseña.....	38
Figura 31: Envío Correo Electrónico Recuperar Contraseña .....	38
Figura 32: Correo Electrónico Recuperar Contraseña .....	39
Figura 33: Pantalla Introducir Contraseña Recuperar Contraseña .....	39
Figura 34: Pantalla Inicio .....	40
Figura 35: Elección Tipo Gráfico y Fecha .....	40
Figura 36: Gráfico Número de Carros por Día .....	41
Figura 37: Pantalla Avisos.....	41
Figura 38: Pantalla Cámaras en Directo .....	42
Figura 39: Pantalla Perfil .....	42
Figura 40: Pantalla Editar Perfil.....	43
Figura 41: Pantalla Modificar Contraseña.....	43
Figura 42: Pantalla Eliminar Cuenta .....	44
Figura 43: Pantalla Gestión de Usuarios .....	44
Figura 44: Pantalla Editar Usuario .....	45
Figura 45: Pantalla Eliminar Usuario.....	45
Figura 46: Pantalla Gestión de Zonas.....	46
Figura 47: Pantalla Añadir Zona.....	46
Figura 48: Pantalla Editar Zona.....	47
Figura 49: Pantalla Eliminar Zona .....	47
Figura 50: Pantalla Gestión de Cámaras .....	48
Figura 51: Pantalla Añadir Cámara 1 .....	48
Figura 52: Pantalla Añadir Cámara 2 .....	49
Figura 53: Pantalla Editar Cámara 1 .....	49

Figura 54: Pantalla Editar Cámara 2 .....	50
Figura 55: Pantalla Eliminar Cámara .....	50
Figura 56: Pantalla Gestión de Avisos .....	51
Figura 57: Pantalla Añadir Aviso .....	51
Figura 58: Pantalla Editar Aviso .....	52
Figura 59: Pantalla Eliminar Aviso .....	52
Figura 60: Curva Precisión-Recall.....	54

## Índice de Tablas

Tabla 1: Participante Javier Caballero Sandoval .....	19
Tabla 2: Objetivo Gestión de Usuarios .....	20
Tabla 3: Requisito de Información sobre Usuarios .....	21
Tabla 4: Caso de Uso Registrar Usuario .....	23
Tabla 5: Requisito No Funcional Usabilidad .....	24

# 1. Introducción

En la actualidad, es importante recoger datos de los carros de la compra en un supermercado ya que, con estos, se pueden resolver posibles problemas o mejorar aspectos como la seguridad o la experiencia del cliente.

El sistema que se plantea consiste en una plataforma para el análisis de la densidad de los carros de la compra en un supermercado. Está compuesto por un software capaz de localizar y trackear carros de la compra a partir de las cámaras colocadas en el supermercado y una aplicación web para poder visualizar los datos recogidos por el anterior software, así como personalizar diferentes aspectos como las zonas, las cámaras y los avisos.

En este documento se detalla los aspectos más importantes del ciclo de vida del proyecto. Presenta la siguiente estructura:

- **Estado del arte.** Realiza la estimación del esfuerzo y la planificación temporal del proyecto.
- **Objetivos.** Se definen los objetivos que debe cumplir el sistema, así como los objetivos personales a realizar.
- **Conceptos teóricos.** Se explican los conceptos teóricos que son necesarios para la comprensión del sistema.
- **Técnicas y herramientas.** Se detallan las técnicas y herramientas utilizadas para desarrollar el sistema.
- **Aspectos relevantes del desarrollo.** Se explican los aspectos más importantes en el desarrollo del sistema.
- **Limitaciones del sistema.** Se exponen las limitaciones encontradas durante la realización del sistema.
- **Conclusiones y líneas de trabajo futuras.** Se explican las conclusiones obtenidas tras la realización del sistema, así como las posibles mejoras del sistema que se realizarán en el futuro.
- **Bibliografía.**

La memoria es complementada por los siguientes anexos:

- **Anexo I Planificación Temporal.** Realiza la estimación del esfuerzo y la planificación temporal del proyecto.
- **Anexo II Especificación de Requisitos.** Documentación de la fase de especificación de los requisitos.
- **Anexo III Análisis de Requisitos.** Documentación de la fase de análisis de los requisitos.

- **Anexo IV Diseño del Sistema Software.** Documentación de la fase de diseño del sistema.
- **Anexo V Documentación Técnica.** Detalla el código implementando para su comprensión.
- **Anexo VI Manual de Usuario.** Detalla la funcionalidad del sistema para facilitar su uso.

## 2. Estudio del Estado del Arte

### 2.1. Investigación

En el inicio del proyecto se ha llevado a cabo una investigación acerca de sistemas cuyo objetivo sea similar a este proyecto permitiendo conocer diferentes puntos de vista acerca del campo y orientar de manera correcta el proyecto.

Se ha encontrado un estudio que trata de detectar los carros perdidos en el aparcamiento del supermercado y no devueltos a su zona de estacionamiento estándar. Esto permite resolver posibles problemas como el carro se encuentra en una zona de tránsito de vehículos y puede provocar un accidente, así como evitar la ausencia de carros en el estacionamiento estándar debido a que numerosos carros se encuentran en diferentes puntos del aparcamiento sin ser encontrados [1].

Otro estudio encontrado busca detectar y trackear a las personas durante su estancia en una cola de un servicio del supermercado. Esto permite conocer los patrones en los que se producen aglomeraciones en un punto concreto del supermercado generando insatisfacción del cliente y estudiar los casos para llevar a cabo una posible solución [2].

Finalmente, otro estudio, cuyo objetivo es más similar al de este proyecto que los anteriores, realiza el trackeo a través de incluir un dispositivo en cada carro de la compra e instalar diferentes balizas a lo largo del espacio a trackear para conocer en tiempo real la posición exacta de cada carro [3].

Por otra parte, se ha realizado una investigación acerca de los diferentes algoritmos de detección de objetos en tiempo real para determinar cuál era el mejor para utilizar en el proyecto. Existen dos ramas en la detección de objetos: los algoritmos que localizan los objetos en dos etapas como por ejemplo RCNN [4] y los que lo llevan a cabo en una única etapa como YOLO [5].

Por un lado, los que utilizan dos etapas, la primera es utilizada para generar una serie de posibles localizaciones para los objetos y en la segunda refinan estas estimaciones y realizan la predicción final. Por otro lado, los algoritmos que utilizan una única etapa procesan la imagen una única vez.

Esto conlleva que los algoritmos de que realizan dos etapas sean más precisos en la localización de objetos, pero son más costosos computacionalmente que los algoritmos de una etapa. Debido a esto, en el proyecto se ha utilizado un algoritmo de una única etapa, YOLO, en su versión 5, ya que en las imágenes de un supermercado se espera localizar numerosos carros en cada imagen siendo la rapidez importante y no es necesario contemplar una localización estricta en cada instante.

### 2.2. Soluciones Comerciales

Tras localizar y trackear a los carros por las diferentes zonas del supermercado se obtienen datos de estos que permiten extraer conclusiones tras analizarlos correctamente.

Las conclusiones que se obtienen están relacionadas con los hábitos de consumo de los clientes como por ejemplo tendencias sociales, días o horarios de aumento y decremento de número de clientes.

El uso del análisis de los datos permite conocer el estado del supermercado en tiempo real lo que ayuda a tomar decisiones empresariales respecto al stock, promociones o precios de los productos permitiendo una mayor ganancia económica para el supermercado y mejora en la experiencia del cliente.

Existen numerosos ejemplos de softwares que implementan la funcionalidad de Business Intelligence que permiten la recogida y el tratamiento de los datos obtenidos en un supermercado.

Un ejemplo es la aplicación, enfocada en los supermercados, desarrollada por Distrito K, además de proporcionar funcionalidad relativa a la gestión de un supermercado como la compra y venta de productos, gestión de fidelización del cliente o gestión de tesorería aporta la funcionalidad de Business Intelligence que permite visualizar los datos recogidos del supermercado para el posterior análisis [6].



Figura 1: Distrito K [6]

Otro ejemplo es el software desarrollado por Wyn Enterprise para el sector retail. De la misma manera que el software mencionado anteriormente proporciona diferentes servicios para gestionar las diferentes secciones que forman un supermercado y adicionalmente una funcionalidad para la visualización y análisis de los datos recogidos [7].



Figura 2: Wyn Enterprise [7]

## 3. Objetivos

En esta sección se definen los objetivos que debe tener el sistema a desarrollar en el Trabajo Fin de Grado. Se diferencian los objetivos del sistema y los personales.

### 3.1. Objetivos del Sistema

El objetivo principal del sistema es la localización y trackeo de carros de la compra a partir de las imágenes proporcionadas por una cámara y posteriormente visualizar los datos recogidos. Los objetivos definidos son:

- **Gestión de usuarios.** El sistema debe permitir crear, modificar, visualizar y eliminar usuarios del sistema.
- **Gestión de carros.** El sistema tiene la capacidad de reconocer carros de la compra a través de imágenes y trackearlos. Además, debe almacenar información acerca de estos.
- **Gestión de zonas.** El sistema debe permitir crear, modificar, visualizar y eliminar zonas del sistema.
- **Gestión de cámaras.** El sistema debe permitir crear, modificar, visualizar y eliminar cámaras del sistema.
- **Gestión de avisos.** El sistema debe permitir crear, modificar, visualizar y eliminar avisos del sistema. Además, debe mostrar los avisos detectados por las cámaras.
- **Gestión de estadísticas.** El sistema debe mostrar diferentes estadísticas sobre los carros para visualizar de manera gráfica distintos comportamientos sobre los carros en el supermercado.

### 3.2. Objetivos Personales

En este apartado se exponen los objetivos personales que se intentan cumplir tras realizar el proyecto y los motivos a realizarlo.

Los últimos años, debido al aumento de avances en la inteligencia artificial, tenía curiosidad acerca de este campo, en concreto en la detección de objetos en tiempo real y el Trabajo Fin de Grado era una oportunidad para llevar a cabo un proyecto sobre este tema. El campo de la inteligencia artificial está asociado al avance de la sociedad y personalmente quiero realizar un proyecto de este tema para conocerlo más a fondo para que en el futuro tenga más conocimientos y pueda desarrollar otros sistemas más avanzados para facilitar o ayudar en algunas tareas a la sociedad.

Además, este proyecto abarca conocimientos abordados en el grado, pero también ha sido necesaria una amplia investigación para conocer los detalles de los conceptos que desconocía lo que ha generado un gran interés en el campo de la investigación tras no haberla tratada previamente.

Por otro lado, desde el comienzo del proyecto he presentado una gran motivación debido a tener la posibilidad de llevar a cabo la realización de un sistema capaz de ser utilizado en el ámbito comercial en numerosos establecimientos facilitando la capacidad de recopilar información acerca de los carros de la compra.

Por último, es un objetivo personal verificar la capacidad personal a la hora de realizar un proyecto completo por primera vez tras realizar numerosos trabajos simples de aprendizaje durante los cursos del grado.

## 4. Conceptos Teóricos

En esta sección se detallan los conceptos teóricos utilizados en el desarrollo del sistema para poder comprenderlo mejor.

### 4.1. Detección de Objetos

La detección de objetos es una técnica de visión computacional que permite la localización de objetos a partir de imágenes o videos. Los algoritmos de detección de objetos utilizan machine learning o deep learning para generar resultados satisfactorios.

La detección de objetos se utiliza, en la actualidad, en números ámbitos como por ejemplo la conducción automática o el médico.

### 4.2. Trackeo de Objetos

El trackeo de objetos es una aplicación de la visión computacional en el que, previamente localizado el objeto, se sigue los movimientos de dicho objeto en el espacio observado o en diferentes ángulos de la cámara. El trackeo de objetos permite seguir a diferentes objetos en una imagen.

Una de las utilidades en la actualidad es la estimación de la posición en tiempo real para los objetos en movimiento en el campo de la realidad aumentada.

### 4.3. Rest

Es una arquitectura utilizada para crear aplicar aplicaciones web. Aporta una serie de restricciones para crearlas. Tiene las siguientes características:

- Se basa en una arquitectura cliente-servidor que se comunica a través del protocolo HTTP.
- Comunicación sin estado, el servidor no almacena información del estado del cliente.
- Peticiones cacheables, evitando de esta manera repetir conexiones entre cliente y servidor.
- Interfaz uniforme entre los diferentes componentes.
- Sistema de capas organizadas [8].

## 5. Técnicas y Herramientas

En esta sección se detallan las técnicas, herramientas, bibliotecas y módulos software utilizados a lo largo del ciclo de vida del proyecto. Se dividen en:

- Tracker
- Aplicación Web.
- Servidor.
- Editor de Texto.
- Herramientas CASE.

### 5.1. Tracker

#### 5.1.1. YOLO v5

Es un modelo de la familia de visión computacional YOLO (You Only Look Once). Es un algoritmo que emplea una única red neuronal convolucional (CNN) que permite detectar objetos en tiempo real a partir de imágenes.

YOLO es más rápido que otros sistemas de detección de objetos debido a que analiza la imagen entera en vez de píxel a píxel. Esto lo lleva a cabo dividiendo la imagen en un grid y cada sección del grid clasifica y localiza los objetos en ella. Las predicciones se llevan a cabo a través de un algoritmo basado en regresión en vez de en un algoritmo basado en la clasificación, que realizan dos pasos [9].

#### 5.1.2. Roboflow

Es una herramienta utilizada en aplicaciones de visión computacional. Tiene diferentes utilidades como etiquetar los objetos en una imagen, preprocesar imágenes, redimensionar, cambiar orientación, ajustar contraste, entrenar modelos u obtener o publicar datasets [10].

En este proyecto se ha utilizado para etiquetar los carros de la compra en diferentes imágenes, como se muestra en la [Figura 3](#), generando el dataset que permite entrenar al modelo YOLO v5.

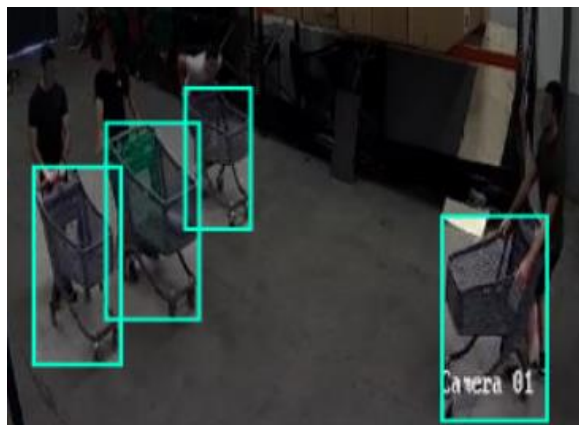


Figura 3: Roboflow

### 5.1.3. Google Colab

Es un producto de Google que permite escribir y ejecutar código Python en el navegador. Es un servicio de notebooks de Jupyter y permite utilizar recursos computacionales como CPU o GPU de manera gratuita [\[11\]](#).

### 5.1.4. Python

Python es un lenguaje de programación orientado a objetos de propósito general que tiene numerosas aplicaciones en diferentes aspectos del software. Su semántica dinámica, sus estructuras de datos de alto nivel, su tipado y su vinculación dinámicos lo convierten en uno de los lenguajes más útiles para el desarrollo rápido de aplicaciones.

Python es normalmente usado en análisis de datos, visualización de los datos y desarrollo web. Lleva a cabo un papel muy importante en la ciencia de datos debido a que es utilizado para ejecutar cálculos complejos y crear algoritmos de inteligencia artificial. Se emplea Python para estas tareas debido a que es más potente que otros lenguajes de programación porque es un lenguaje de programación interpretado, es decir, las instrucciones se ejecutan sin necesidad de ser compiladas previamente [\[12\]](#).

### 5.1.5. Flask

Flask es un framework web, un módulo de Python que permite desarrollar aplicaciones web de manera sencilla. Posee un núcleo sencillo y fácil de extender. Flask está basado en la herramienta Werkzeug WSGI y en el motor de plantillas Jinja2 [\[13\]](#).

### 5.1.6. OpenCV

Es una librería software libre de visión computacional y machine learning. OpenCV fue construida para crear una infraestructura común en el campo de la visión computacional y de esta manera acelerar la percepción de las máquinas en los productos comerciales [\[14\]](#).

### 5.1.7. NumPy

Es una librería de Python que provee objetos multidimensionales y numerosos métodos para operar de manera eficiente con matrices. Es utilizada principalmente en ciencia de datos debido a su rendimiento [\[15\]](#).

### 5.1.8. Dlib

Es una librería de propósito general multiplataforma escrita en C++ que contiene algoritmos de machine learning y herramientas para crear software complejo para resolver problemas. Es utilizado en la industria en campos como la robótica, teléfonos móviles y entornos de alto rendimiento [16].

## 5.2. Aplicación Web

### 5.2.1. JavaScript

JavaScript es un lenguaje de alto nivel basado en el estándar ECMAScript. Es utilizado en el desarrollo web aportando la funcionalidad.

Presenta las siguientes características: débilmente tipado y dinámico, programación orientada a objetos, interpretado, multiparadigma, programación dirigida por eventos e imperativa [17].

### 5.2.2. HTML

HTML (Hypertext Markup Language) es el lenguaje de marcado para la creación de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web [18].

### 5.2.3. CSS

CSS (Cascading Style Sheets) es un lenguaje de estilo utilizado para proporcionar la presentación de un documento escrito en un lenguaje de marcado. Se utiliza en el diseño visual de las interfaces de las páginas web, normalmente escritas en HTML [19].

### 5.2.4. Vue.js

Vue.js es un framework de JavaScript que permite crear interfaces de usuario de manera sencilla.

Utiliza HTML, CSS y JavaScript y proporciona un modelo de programación declarativo y basado en componentes lo que permite reutilizarlos y desarrollar las interfaces de usuario de la aplicación de manera más eficiente.

Un componente está dividido en tres partes:

- **<template></template>**. Se implementa el código que proporciona la estructura del componente. Utiliza HTML.
- **<script></script>**. En esta etiqueta se implementa la funcionalidad del componente a través del lenguaje de programación JavaScript.

- `<style></style>`. Se encuentra el diseño del componente utilizando CSS [\[20\]](#).

### 5.2.5. Vuetify

Vuetify es una librería que contiene una colección de componentes de interfaz de usuario ya construidos para el framework Vue.js permitiendo la creación de una aplicación web de manera sencilla y rápida [\[21\]](#).

### 5.2.6. PrimeVue

Es una librería que permite utilizar componentes ya creados de interfaz gráfica de manera sencilla en el framework Vue.js [\[22\]](#).

### 5.2.7. Axios

Axios es un módulo que permite enviar y recibir peticiones basadas en promesas desde el navegador y el servidor. En el servidor utiliza el módulo http nativo de Node.js y en el navegador utiliza XMLHttpRequests [\[23\]](#).

### 5.2.8. Vue-chartjs

Es un módulo que permite crear diferentes tipos de gráficos y es una adaptación de la librería Chart.js para ser utilizada en Vue.js. Además, permite crear componentes de gráficos reutilizables [\[24\]](#).

## 5.3. Servidor

### 5.3.1. NodeJS

Node.js es un entorno de ejecución de ejecución multiplataforma, del lenguaje de programación JavaScript, orientado a eventos asíncronos. Funciona como un modelo de evaluación de un único hilo de ejecución con entradas y salidas asíncronas, que se pueden ejecutar en paralelo sin el coste de cambiar el contexto, siendo una ventaja significativa respecto a los entornos de ejecución basados en hilos [\[25\]](#).

Se ha decidido utilizar este entorno de ejecución debido a que Node.js ofrece numerosos módulos cuya funcionalidad está probada y su rendimiento debido a que se realizan un número significativo de llamadas si el número de usuarios simultáneos es elevado.

### 5.3.2. NPM

NPM (Node Package Manager) es un repositorio de herramientas libre usado para desarrollar aplicaciones y sitios web. Es el sistema de gestión de paquetes por defecto para Node.js. Permite interactuar de manera sencilla con el repositorio a través de la terminal [\[26\]](#).

### 5.3.3. Express

Express es un framework que permite crear aplicaciones web y APIs mediante Node.js. Es una capa construida sobre Node.js que permite manejar servidores y rutas de manera sencilla y rápida [\[27\]](#).

### 5.3.4. MySQL

MySQL es un sistema de gestión de base de datos, permite acceder a los datos de una base de datos. Las bases de datos de MySQL son relacionales, es decir, almacena la información en diferentes tablas relacionadas entre sí en vez de incluir toda la información en una única tabla. El software que ofrece MySQL es libre lo que permite que cualquier desarrollador pueda usar y modificarlo [\[28\]](#).

### 5.3.5. Sequelize

Sequelize es un ORM (Object Relational Mapper) para Node.js. Un ORM es un modelo de programación que convierte los datos de una base de datos relacional en objetos de programación que pueden ser manipulados en un lenguaje de programación orientado a objetos [\[29\]](#).

## 5.4. Editor de Texto

### 5.4.1. Visual Studio Code

Es un editor de texto de código fuente desarrollado por Microsoft. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis y finalización inteligente de código. Además, es posible utilizar módulos de terceros para facilitar alguna de las tareas a realizar. Da soporte a una gran cantidad de lenguajes de programación, entre ellos Python y JavaScript por lo que se ha utilizado como editor de texto en el proceso de implementación en todas las partes del sistema [\[30\]](#).

## 5.5. Herramientas Case

### 5.5.1. Microsoft Project

Es un software de gestión de proyectos que es utilizado para crear calendarios, planes de proyectos y gestionar los recursos. Esta herramienta se ha utilizado para la planificación temporal del proyecto [31].

### 5.5.2. Visual Paradigm

Visual Paradigm es un software que permite crear los diferentes diagramas UML necesarios durante el ciclo de vida de desarrollo del software. Se ha utilizado durante las fases de elicitación de requisitos, diagramas de casos de uso, análisis de requisitos, diagrama de clases y diagramas de secuencia y en el diseño del sistema, diagramas de clase, diagramas de secuencia, diagrama de base de datos y diagrama de despliegue.

### 5.5.3. EZestimate

Es una herramienta utilizada la estimación del esfuerzo permitiendo calcular una aproximación de las horas que durará un proyecto. Esto lo lleva a cabo utilizando los puntos de caso de uso (UCP) y el factor de del número de horas por persona por UCP.

## 5.6. Herramientas de Generación de Documentación

### 5.6.1. Sphinx

Es un generador de documentación para el lenguaje de programación Python, utiliza los comentarios escritos en los ficheros fuentes para generar ficheros en HTML u otros formatos. Se ha utilizado para la documentación relacionada con el Tracker.

### 5.6.2. JSDoc

Es un generador de documentación para el lenguaje de programación JavaScript, utiliza los comentarios escritos en los ficheros fuentes para generar ficheros en HTML. Se ha utilizado para la documentación relacionada con el servidor.

### 5.6.3. Vue Styleguidist

Es un generador de documentación para el framework Vue.js permitiendo la documentación de los componentes, utiliza los comentarios escritos en los ficheros fuentes para generar ficheros en HTML. Se ha utilizado para la documentación relacionada con la aplicación web.

## 6. Aspectos Relevantes del Desarrollo

En esta sección se detallan las partes más relevantes de las distintas fases del desarrollo del proyecto.

### 6.1. Marco de Trabajo

El desarrollo del proyecto ha seguido el Proceso Unificado ya que tiene gran flexibilidad para adaptarse a una gran variedad de sistemas software y está basado en componentes. Las características principales del Proceso Unificado son:

- Conducido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

El Proceso Unificado se divide en fases, a su vez, cada una de ellas tiene diferentes disciplinas y cada fase se puede realizar varias iteraciones. Las fases que lo componen son:

- **Inicio.** Se desarrollan los casos de negocio y se define el alcance del proyecto.
- **Elaboración.** Se planifica el proyecto, estimación del esfuerzo y planificación temporal. Además, se especifica la mayor parte de los casos de uso del sistema y se diseña la arquitectura.
- **Construcción.** Se realiza la construcción del producto.
- **Transición.** El proyecto se convierte en una primera versión que se corregirán los problemas y se añaden las sugerencias aportadas.

En la [Figura 4](#) se observan las diferentes fases con sus disciplinas, así como su carga en el proyecto a lo largo del ciclo de vida.

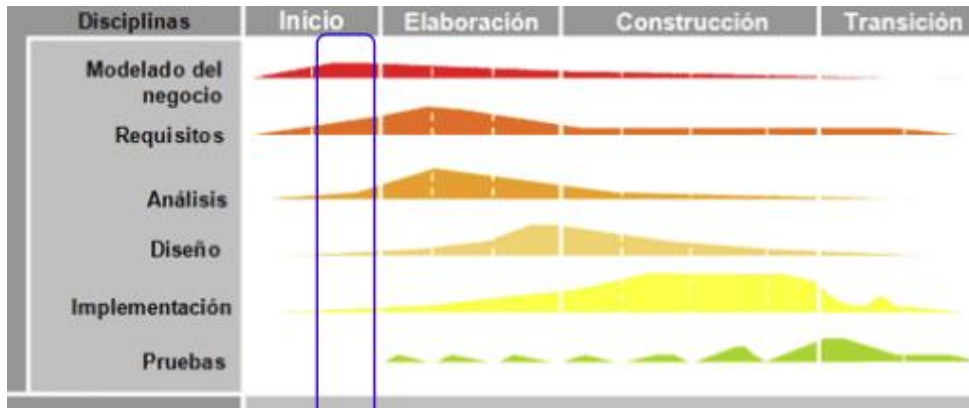


Figura 4: Diagrama de Proceso Unificado [32]

## 6.2. Estimación de la Duración del Proyecto

La estimación de la duración del proyecto se realiza tras definir los casos de uso ya que para su cálculo son necesarios. Tras obtener dicha estimación es posible planificar el tiempo dedicado para cada una de las tareas del proyecto en el tiempo.

Para obtener más información acerca de la estimación de la duración puede consultar el *Anexo I Plan del Proyecto*. Los parámetros introducidos y la estimación obtenida se pueden observar en las siguientes figuras.

**Module**

Autenticación de Usuarios

Add Module Delete

**Summary**

Total Modules: 7 **Excel Report** Generate Report

Use cases: Simple 37 Average 1 Complex 0

Actors: Simple 1 Average 1 Complex 3

**Add Actor / Use case**

Actor / Use case Name:  Select Type:  Complexity:  Add

**Tech / Env Factors**

Set Tech Factor

Set Env Factors

**Estimation Summary**

UAW: 12

UUCW: 195

UUPC = UAW + UUCW: 207

TFactor: 23

EFactor: 17

TCF = 0.6 + (.01\*TFactor): 0.83

EF = 1.4 + (-0.03\*EFactor): 0.89

UCP = UUPC\*TCT\*EF: 152.9109

Total Effort@ 4 Hrs/UCP: 611.6436

**Use case / Actor List** (Double click to delete)

Id	Module	Type	Name	complexity
1	Autenticación d...	Actor	Usuario No Log...	Complex
10	Autenticación d...	Usecase	Cerrar Sesión	Simple
11	Gestión de Usu...	Usecase	Ver datos Usuario	Simple
12	Gestión de Usu...	Usecase	Modificar Usuario	Simple
13	Gestión de Usu...	Usecase	Modificar Contr...	Simple
14	Gestión de Usu...	Usecase	Eliminar Usuario...	Simple
15	Gestión de Usu...	Usecase	Listar Usuarios	Simple
16	Gestión de Usu...	Usecase	Filtrar Usuarios	Simple
17	Gestión de Usu...	Usecase	Modificar Rol	Simple
18	Gestión de Usu...	Usecase	Eliminar Usuario	Simple
19	Gestión de Carros	Usecase	Consultar Strea...	Simple
2	Autenticación d...	Actor	Usuario	Complex
20	Gestión de Carros	Usecase	Registrar Carro	Simple
21	Gestión de Carros	Usecase	Trackear Carro	Simple
22	Gestión de Carros	Usecase	Borrar Carro	Simple
23	Gestión de Carros	Usecase	Calcular Tiemp...	Simple
24	Gestión de Carros	Usecase	Detectar Aviso	Simple
25	Gestión de Zonas	Usecase	Listar Zonas	Simple

Figura 5: Estimación del Esfuerzo

**Technical complexity factors**

Factor	Relevance
Distributed system	1
Response / Throughput performance objectives	2
End-user efficiency	1
Complex internal processing	3
Reusable code	2
Easy to install	1
Easy to use	2
Portable	1
Easy to change	3
Concurrent	3
Includes security features	2
Third party access	0
Special user training facilities required	0

OK Cancel

Figura 6: Factores de Complejidad Técnica

**Environmental factors**

Factor	Relevance
Familiar with Rational unified process	2
Application experience	0
Object oriented experience	2
Lead analyst capability	0
Motivation	5
Stable requirements	4
Part-time workers	0
Difficult programming language	1

OK Cancel

Figura 7: Factores de Entorno

### 6.3. Planificación Temporal

La planificación temporal se realiza tras obtener la estimación de la duración del desarrollo del proyecto. A través de la planificación temporal se identifican las tareas a realizar a lo largo del ciclo de vida del sistema, así como los recursos y tiempo que es necesario dedicar a cada una de ellas. Se ha utilizado el Proceso Unificado durante el desarrollo del sistema por lo que se han organizado las tareas en cada una de las disciplinas de las fases.

- **Modelo de negocio.** Se lleva a cabo la investigación y análisis acerca de los recursos necesarios para realizar el proyecto.
- **Requisitos.** Se fijan los objetivos los requisitos que debe tener el proyecto a realizar.
- **Análisis.** Se realiza el análisis de los requisitos previamente declarados.
- **Diseño.** Se realiza la definición de cómo funcionan y como son los diferentes componentes del sistema.
- **Implementación.** Se lleva a cabo la creación de los programas del sistema.
- **Pruebas.** Se realizan pruebas unitarias y completas del sistema para comprobar que el sistema funciona correctamente y corregir los posibles fallos.

En la [Figura 8](#) se muestran algunas de las tareas a realizar junto a su diagrama de Gantt.



Figura 8: Planificación Temporal

Para obtener más información acerca de la estimación de la duración puede consultar el *Anexo I Plan del Proyecto*.

## 6.4. Especificación de Requisitos

En este apartado se recogen las especificaciones de los requisitos software del sistema a desarrollar. Para ello, se sigue se sigue la metodología de Durán y Bernárdez. Las siguientes fases del ciclo de vida del sistema se basan en esta especificación debido a que contiene las necesidades del proyecto.

Para obtener más información acerca de la especificación de requisitos puede consultar el *Anexo II Especificación de requisitos del Sistema*.

En los siguientes apartados se muestran ejemplos de cada etapa de la especificación de requisitos.

### 6.4.1. Participantes

En el desarrollo del proyecto participan los siguientes integrantes:

- Javier Caballero Sandoval.
- Héctor Sánchez San Blas.
- Gabriel Villarrubia González.

En la siguiente tabla se muestra un ejemplo de la especificación de participantes.

<b>Participante</b>	Javier Caballero Sandoval
<b>Organización</b>	<a href="#">Universidad de Salamanca</a>
<b>Rol</b>	Desarrollador
<b>Es desarrollador</b>	Sí
<b>Es cliente</b>	No
<b>Es usuario</b>	No
<b>Comentarios</b>	Ninguno

*Tabla 1: Participante Javier Caballero Sandoval*

### 6.4.2. Objetivos del Sistema

Los objetivos que se han identificado para resolver las necesidades planteadas son:

- Gestión de Usuarios.
- Gestión de Carros.
- Gestión de Zonas.
- Gestión de Cámaras.
- Gestión de Avisos.
- Gestión de Estadísticas.

Un ejemplo de la especificación de objetivos es:

<b>OBJ-001</b>	<b>Gestión de Usuarios</b>
<b>Versión</b>	1.0 (03/07/2023)
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <a href="#">Javier Caballero Sandoval</a></li> </ul>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Héctor Sánchez San Blas</a></li> </ul>
<b>Descripción</b>	<p>El sistema debe administrar la creación de nuevos usuarios, la visualización de los datos personales, la modificación de estos y la eliminación del usuario. En el sistema existen dos roles de usuarios:</p> <ul style="list-style-type: none"> <li>- Usuario: Puede visualizar los datos registrados relativos a los carros y los avisos y ver las cámaras en directo.</li> <li>- Administrador: Además, puede gestionar al resto de usuarios, zonas, cámaras y avisos.</li> </ul>
<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Comentarios</b>	Ninguno

Tabla 2: Objetivo Gestión de Usuarios

### 6.4.3. Requisitos de Información

Los requisitos de información son la información que el sistema debe almacenar para cumplir las necesidades. Los requisitos de información identificados son:

- Información sobre Usuarios.
- Información sobre Carros.
- Información sobre Zonas.
- Información sobre Cámaras.
- Información sobre Avisos.

Un ejemplo de la especificación de requisitos de información es:

<b>IRQ-001</b>	<b>Información sobre Usuarios</b>
<b>Versión</b>	1.0 (03/07/2023)
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <a href="#">Javier Caballero Sandoval</a></li> </ul>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Héctor Sánchez San Blas</a></li> </ul>
<b>Dependencias</b>	<a href="#">[OBJ-001] Gestión de Usuarios</a>
<b>Descripción</b>	El sistema debe almacenar la información acerca de los usuarios registrados en el sistema.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• Nombre</li> <li>• Apellidos</li> <li>• Nombre de usuario</li> </ul>

	<ul style="list-style-type: none"> <li>• Email</li> <li>• Contraseña</li> <li>• Verificado</li> <li>• Rol</li> </ul>
<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

Tabla 3: Requisito de Información sobre Usuarios

#### 6.4.4. Requisitos Funcionales

El sistema se divide en diferentes paquetes, mostrados en la [Figura 9](#), modulando el sistema.

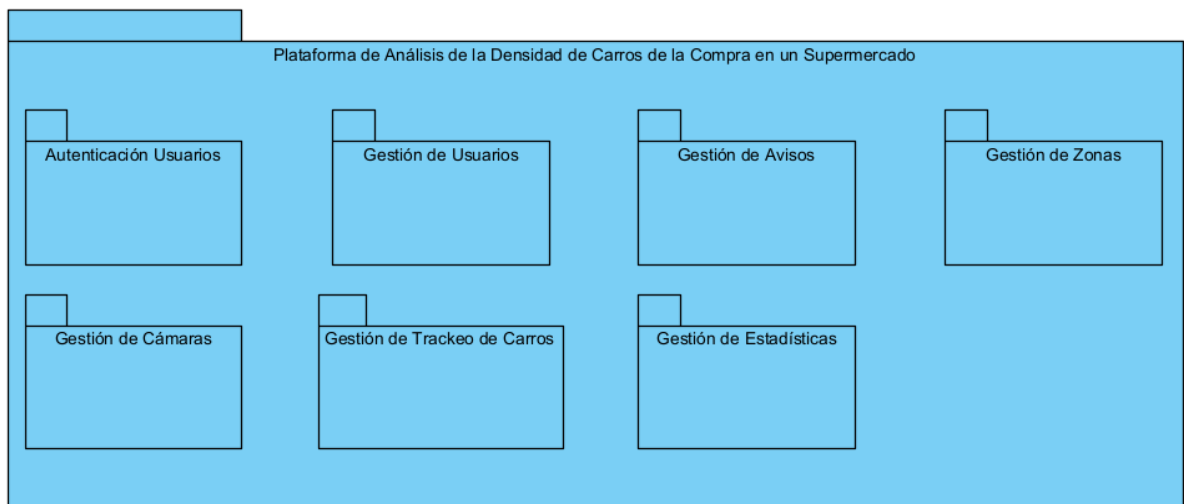


Figura 9: Diagrama de Paquetes

A continuación, se definen los actores que participan en el sistema y su jerarquía.

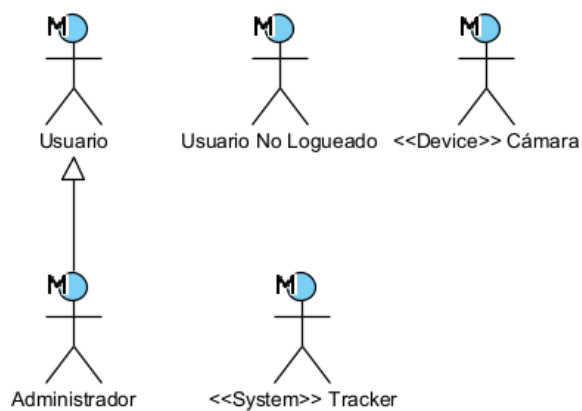


Figura 10: Actores del Sistema

Un ejemplo de la especificación de los actores es:

<b>ACT-001</b>	<b>Usuario No Logueado</b>
<b>Versión</b>	1.0 (03/07/2023)
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <a href="#">Javier Caballero Sandoval</a></li> </ul>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Héctor Sánchez San Blas</a></li> </ul>
<b>Descripción</b>	Este actor representa a un usuario antes de iniciar sesión.
<b>Comentarios</b>	Ninguno

Finalmente, se definen y detallan los casos de uso, que muestran como se comporta el sistema. Los diagramas de casos de uso se han realizado individualmente respecto a los paquetes indicados previamente. Un ejemplo de diagrama de uso es:

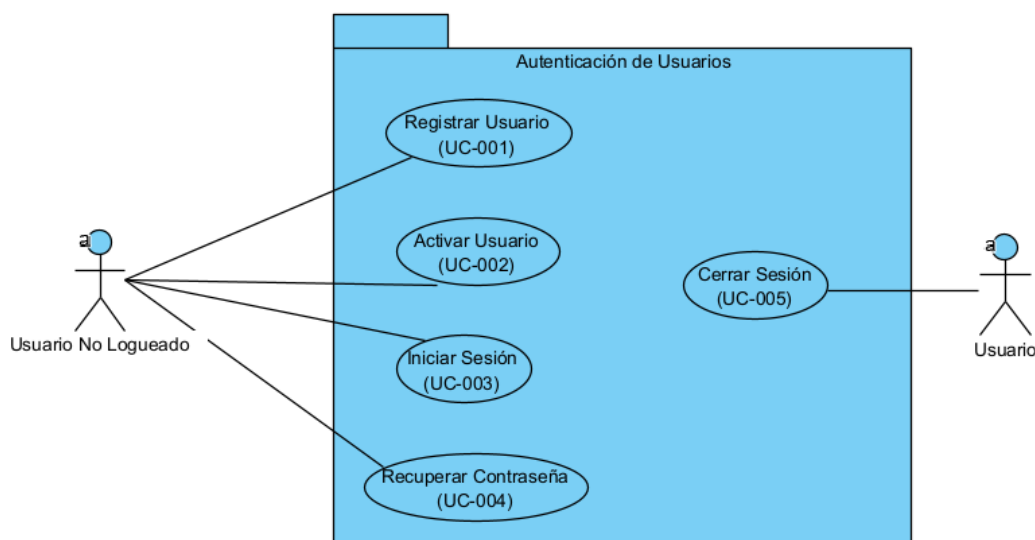


Figura 11: Diagrama de Casos de Uso Autenticación de Usuarios

Los casos de uso se han especificado individualmente como se muestra en el siguiente ejemplo.

<b>UC-001</b>	<b>Registrar Usuario</b>
<b>Versión</b>	1.0 (03/07/2023)
<b>Autores</b>	<ul style="list-style-type: none"> <li>• <a href="#">Javier Caballero Sandoval</a></li> </ul>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Héctor Sánchez San Blas</a></li> </ul>
<b>Dependencias</b>	<a href="#">[OBJ-001] Gestión de Usuarios</a> <a href="#">[IRQ-001] Información sobre Usuarios</a>
<b>Descripción</b>	El sistema solicita al usuario que introduzca sus datos personales y posteriormente se crea su cuenta.
<b>Precondición</b>	El usuario no ha iniciado sesión.

<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El actor <a href="#">Usuario No Logueado (ACT-001)</a> solicita el registro de su cuenta.
	2	El sistema solicita al usuario que introduzca sus datos personales.
	3	El actor <a href="#">Usuario No Logueado (ACT-001)</a> introduce los datos requeridos.
4	El sistema comprueba que los datos introducidos son correctos, manda un email para la verificación de la cuenta y notifica al actor <a href="#">Usuario No Logueado (ACT-001)</a> que debe verificar su usuario a través del correo electrónico introducido.	
<b>Postcondición</b>	El usuario se crea en el sistema con los datos introducidos.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	El actor <a href="#">Usuario No Logueado (ACT-001)</a> cancela el registro, este caso de uso queda sin efecto.
4	Los datos introducidos son incorrectos o los campos nombre de usuario y/o correo electrónico ya están registrados en el sistema, el sistema devuelve un mensaje de error y el caso de uso queda sin efecto.	
<b>Importancia</b>	Alta	
<b>Urgencia</b>	Alta	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno.	

Tabla 4: Caso de Uso Registrar Usuario

#### 6.4.5. Requisitos No Funcionales

Los requisitos de no funcionales imponen restricciones en el diseño e implementación del sistema, así como estándares de calidad. Los requisitos no funcionales identificados son:

- Usabilidad.
- Compatibilidad.
- Fiabilidad.
- Implementación.
- Privacidad.

Un ejemplo de la especificación de requisitos de no funcionales es:

<b>NFR-001</b>	<b>Usabilidad</b>
<b>Versión</b>	1.0 (03/07/2023)
<b>Autores</b>	<a href="#">Javier Caballero Sandoval</a>
<b>Fuentes</b>	<ul style="list-style-type: none"> <li>• <a href="#">Gabriel Villarrubia González</a></li> <li>• <a href="#">Héctor Sánchez San Blas</a></li> </ul>
<b>Dependencias</b>	Ninguna
<b>Descripción</b>	El sistema debe presentar una aplicación web con una interfaz intuitiva y sencilla para que cualquier usuario pueda llevar a cabo sus necesidades fácilmente.

<b>Importancia</b>	Alta
<b>Urgencia</b>	Alta
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

Tabla 5: Requisito No Funcional Usabilidad

## 6.5. Análisis de Requisitos

En esta sección se presenta el análisis orientado a objetos del sistema planteado. Este análisis consiste en una serie de técnicas y actividades mediante las que los requisitos identificados son analizados, refinados y estructurados. Permite comprender de manera más precisa los requisitos.

Para obtener más información acerca de la estimación del análisis de requisitos puede consultar el *Anexo III Análisis de Requisitos*.

### 6.5.1. Modelo de Dominio

El modelo de dominio muestra las clases significativas identificadas del sistema a desarrollar, así como sus atributos y relaciones. Para realizarlo se ha utilizado un diagrama de clases mostrado en la siguiente figura.

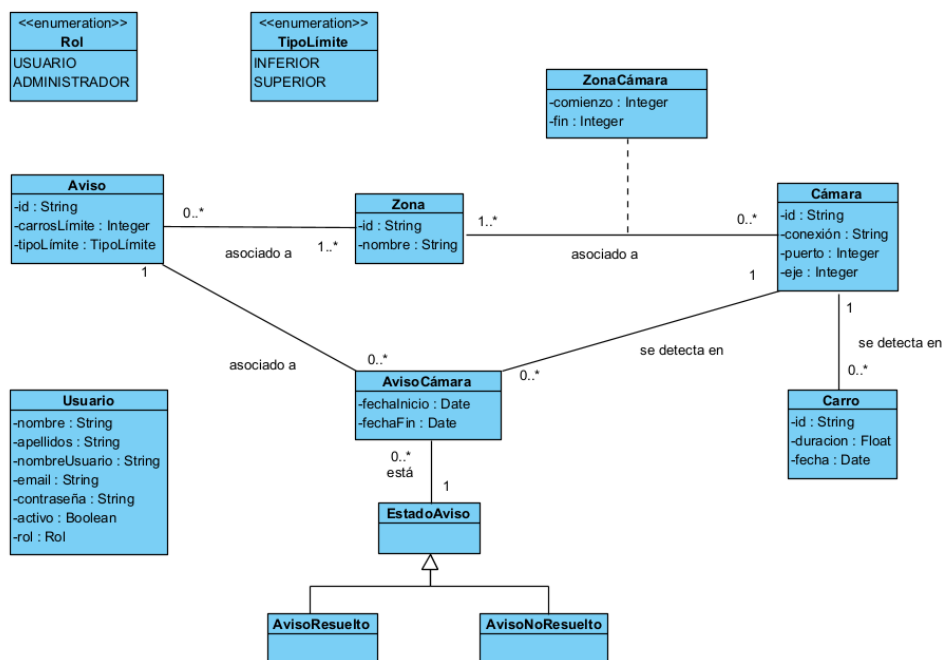


Figura 12: Modelo del Dominio

## 6.5.2. Realización de Casos de Uso Análisis

En la realización de los casos de uso se muestran a través de diagramas de secuencia las interacciones y mensajes de los objetos del sistema. Se muestra un ejemplo de diagrama de secuencia realizado, Registrar Usuario, en la siguiente figura.

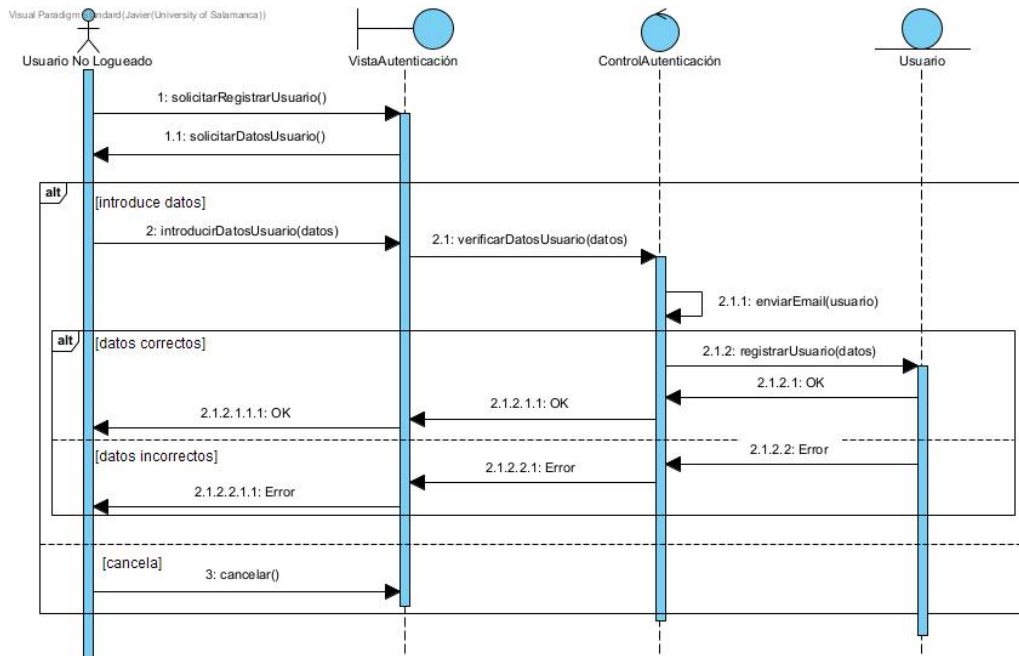


Figura 13: Diagrama de Secuencia Análisis Registrar Usuario

## 6.5.3. Clases de Análisis y Propuesta de Arquitectura

En la [Figura 14](#) se muestran las clases de análisis identificadas, así como la propuesta de arquitectura que presentará el sistema.

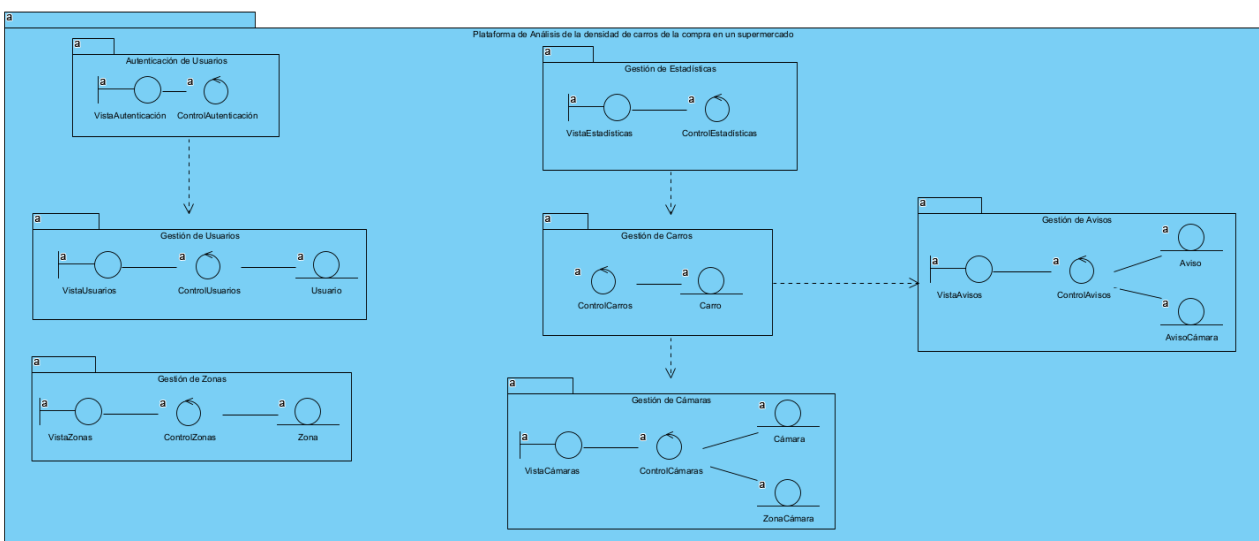


Figura 14: Propuesta de Arquitectura

## 6.6. Diseño del Sistema

En esta sección se presenta el diseño orientado a objetos del sistema planteado. Es un modelo de objetos que describe la realización física de los casos de uso.

Debido a que es una aproximación a la implementación, el nombre de las clases, métodos, y atributos serán un acercamiento a los que se utiliza en la implementación.

Para obtener más información acerca del diseño del sistema puede consultar el *Anexo IV Diseño del Sistema Software*.

### 6.6.1. Patrones Arquitectónicos y de Diseño

En esta sección se detallan los patrones arquitectónicos y de diseño que se utilizarán en el desarrollo del sistema facilitando el mismo teniendo en cuenta las características del sistema.

### 6.6.2. Patrón Modelo -Vista- Modelo de Vista (MVVM)

La aplicación web permite mediante una interfaz recoger y modificar la información del sistema a través de una interfaz. El sistema debe controlar el flujo de la información de manera adecuada, para ello se utiliza el patrón MVVM. La aplicación web se desarrollará utilizando el framework Vue.js que permite crear una aplicación de manera eficiente y sencilla reutilizando componentes. Este framework se basa en el patrón MVVM que permite unir la vista y el modelo de manera bidireccional a través del modelo de vista. El sistema se divide en tres partes:

- **Modelo.** Representa a los datos del sistema.
- **Modelo de Vista.** Aloja el control de la interacción de la vista y conecta los datos del modelo con la vista.
- **Vista.** Presenta los elementos visuales al usuario [\[33\]](#).

## Model-View-ViewModel

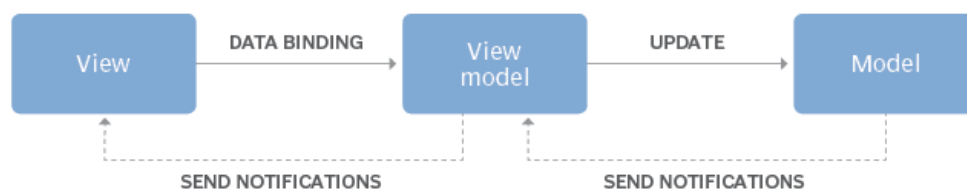


Figura 15: Modelo-Vista-Modelo Vista [\[33\]](#)

### 6.6.3. Patrón DAO

El patrón DAO (Data Access Object) es utilizado para desacoplar la lógica de negocio de la lógica de acceso a datos. En el servidor se lleva a cabo el acceso a los datos de la base de datos, para facilitar su tratamiento se crearán modelos de las entidades a almacenar. Para ello, se utilizará el módulo *Sequelize* que para cada una de las instancias de los modelos previamente creados crea un DAO.

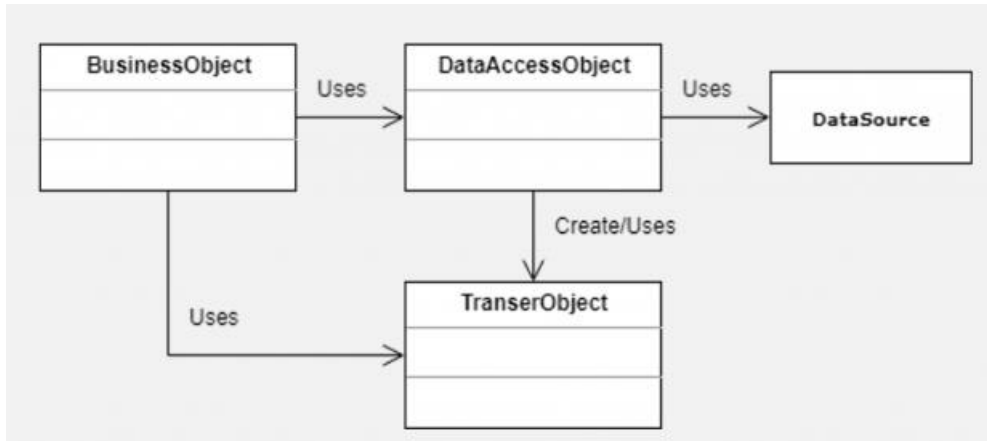


Figura 16: DAO [34]

### 6.6.4. Subsistemas de Diseño

El sistema se ha dividido en diferentes subsistemas facilitando su manejo.

- **Server.** Es el servidor en el que se encuentran los datos persistentes almacenados situados en el paquete Model. Además, se encuentra la especificación del patrón DAO en el paquete DAOApi. En el paquete ControllerApi se implementa la lógica de control de los datos.
- **Web.** Representa la aplicación web con la que interactúa el usuario. Como se ha comentado anteriormente se utilizará el framework Vue.js lo que permite diferenciar los componentes que se reutilizan en diferentes vistas. El Router implementa el cambio de las vistas.
- **Collector.** Representa a los procesos encargados de detectar y trackear los carros de la compra en segundo plano. El servidor de este subsistema permite recuperar las imágenes de las cámaras con el reconocimiento de los carros de la compra.

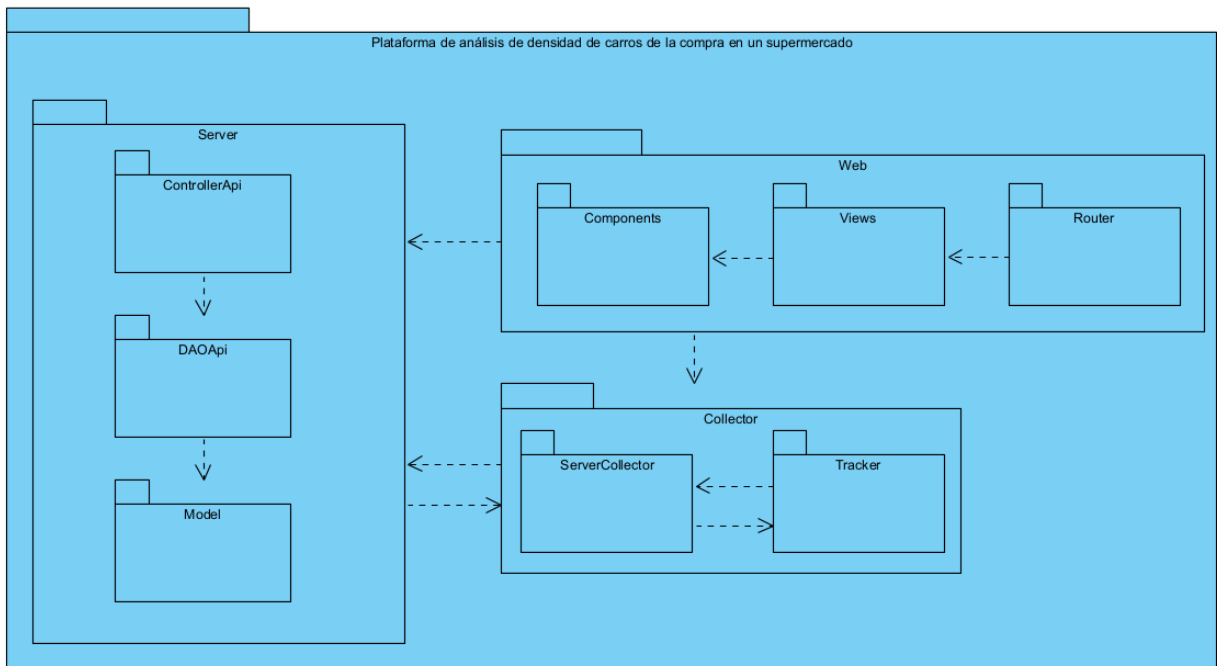


Figura 17: Subsistemas de Diseño

### 6.6.5. Clases de Diseño

En este apartado se especifican las clases de diseño del sistema junto a sus métodos. El servidor presenta un subsistema controlador, un DAOApi y un modelo. Sus clases de diseño se observan en la siguiente figura.

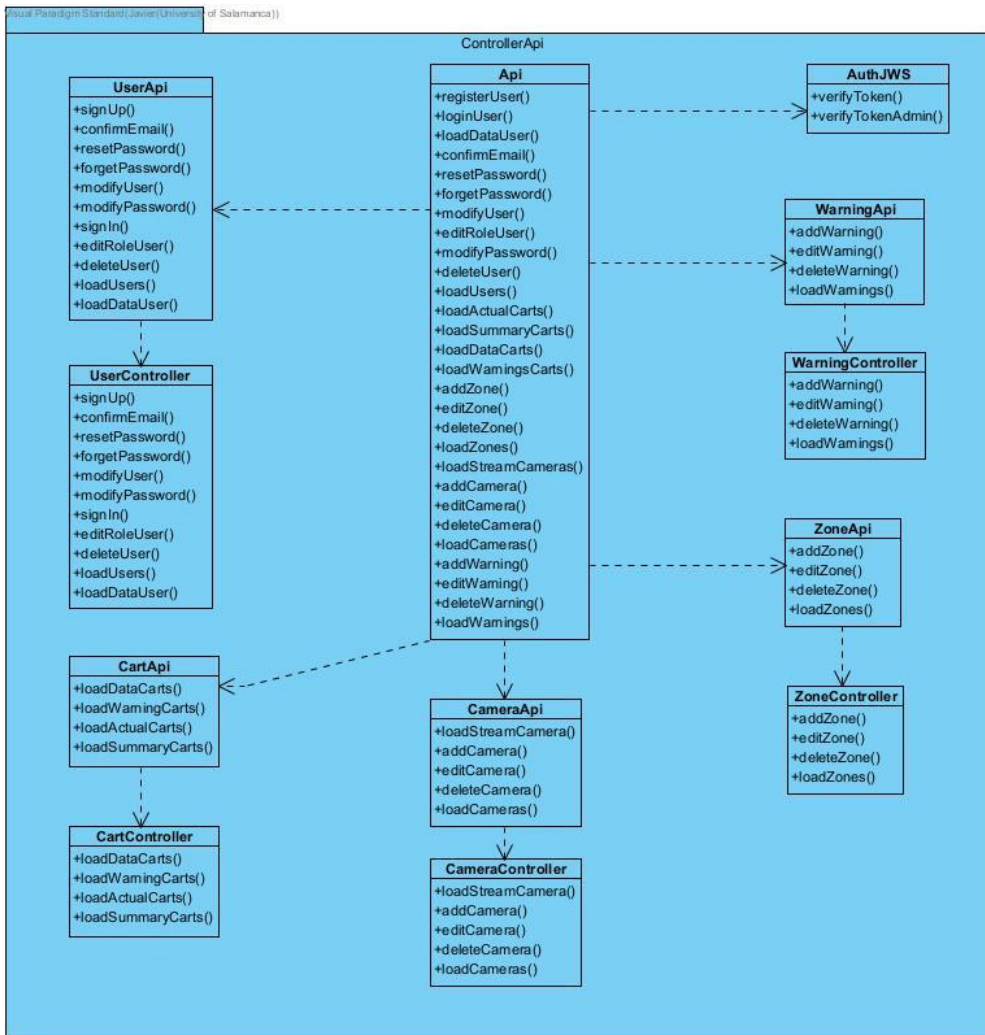


Figura 18: ControllerApi

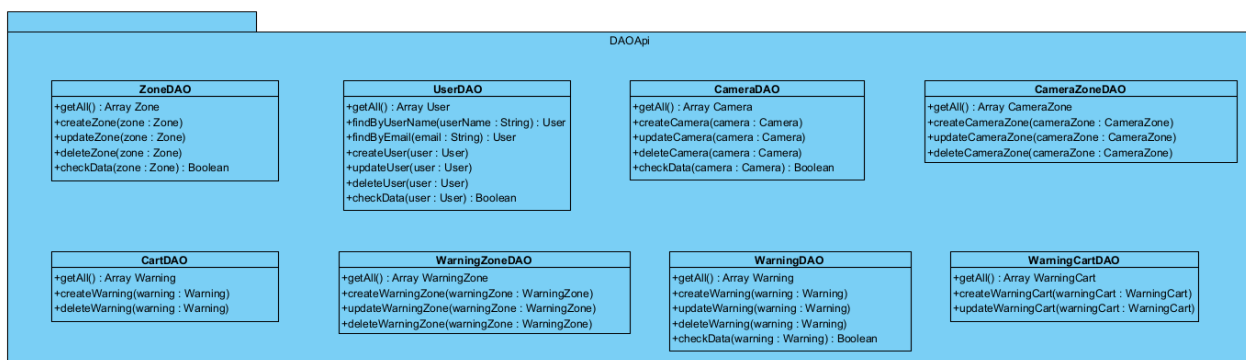


Figura 19: DAOApi

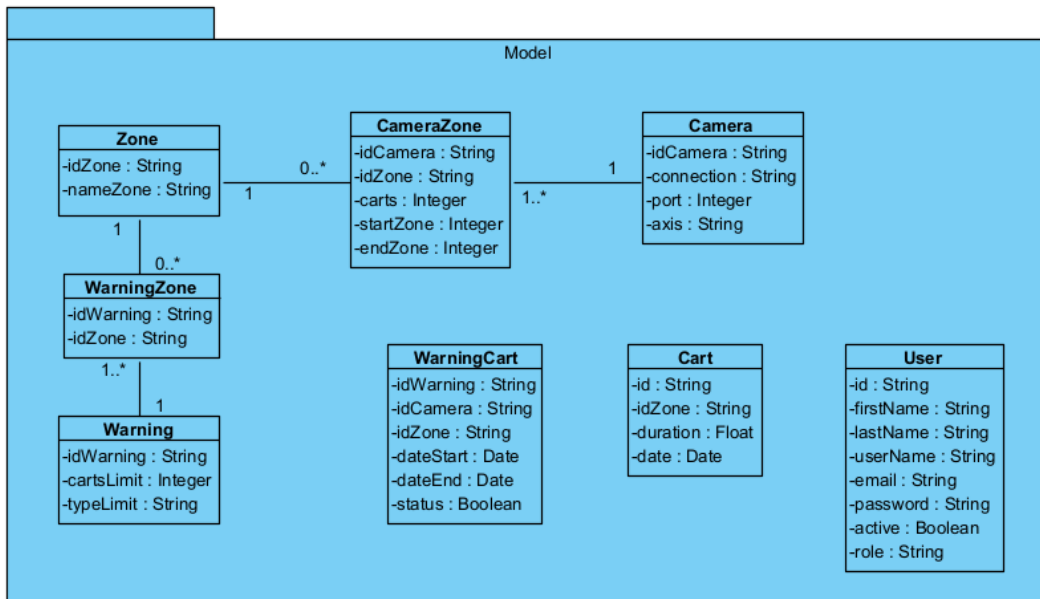


Figura 20: Model

Respecto a la aplicación web, esta se divide en los subsistemas de componentes, vistas y router. Sus clases de diseño son:

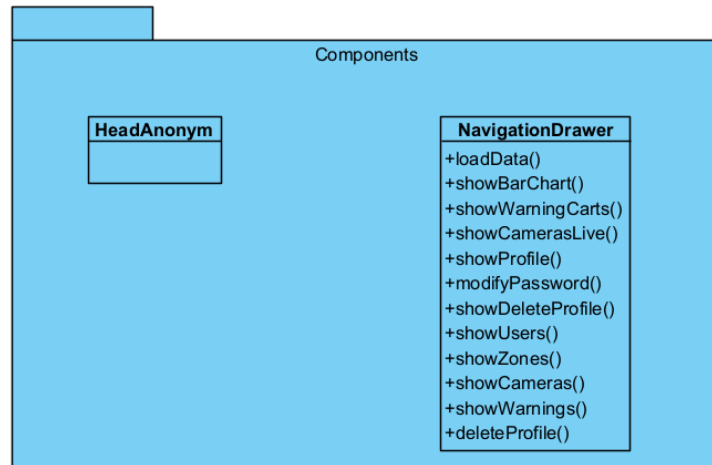


Figura 21: Components

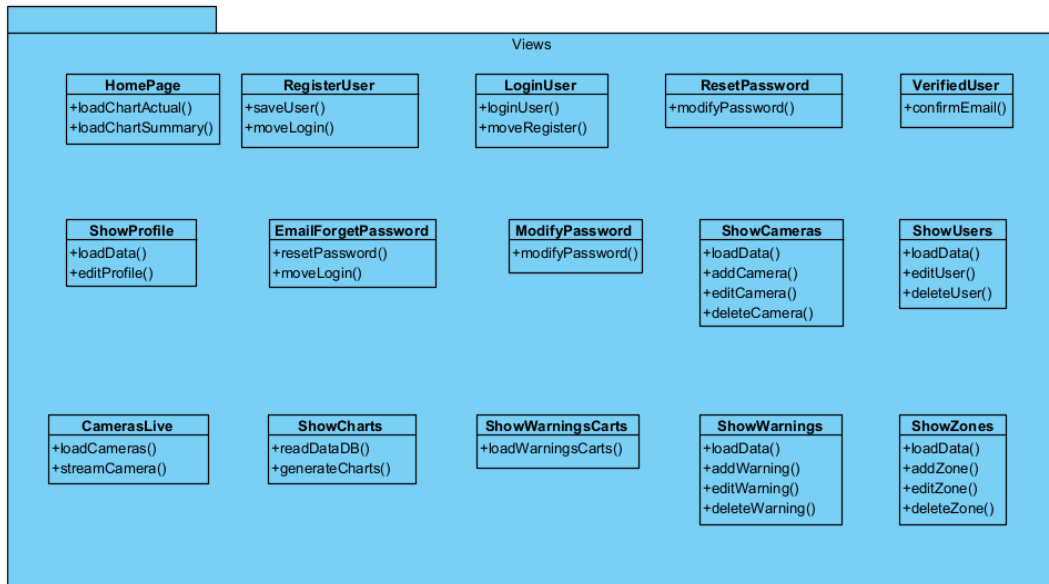


Figura 22: Views

El collector se divide en los subsistemas Tracker y servidor del Tracker. Sus clases de diseño se presentan en las siguientes figuras.

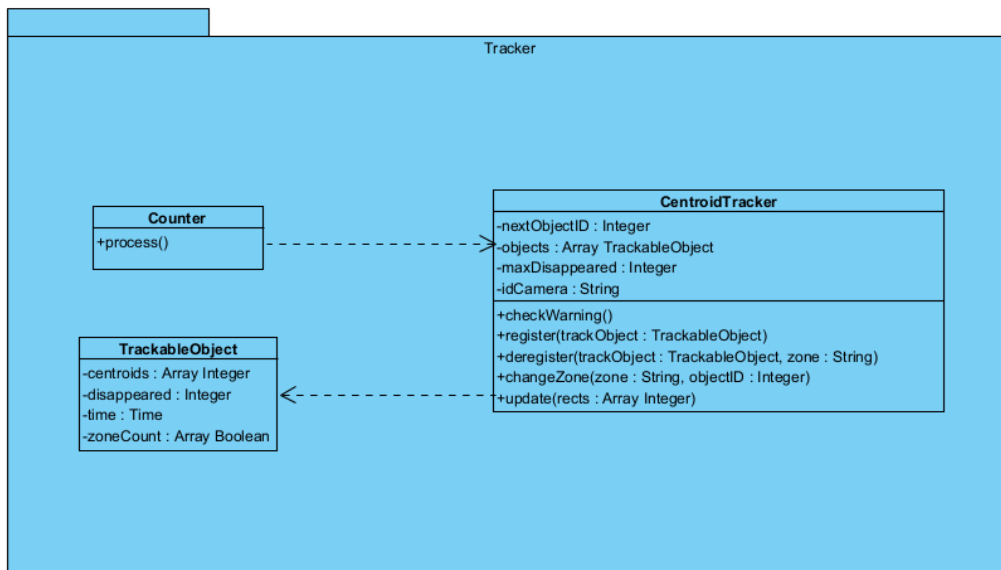


Figura 23: Tracker

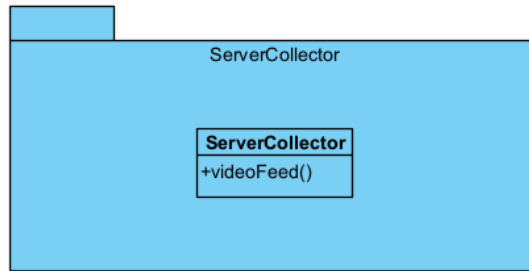


Figura 24: *ServerCollector*

### 6.6.6. Realización de Casos de Uso Diseño

En la realización de los casos de uso se muestran a través de diagramas de secuencia las interacciones y mensajes de los objetos del sistema de manera más detallada que en la realización de casos de uso en la fase de análisis. Se muestra un ejemplo de diagrama de secuencia realizado, Registrar Usuario, en la siguiente figura.

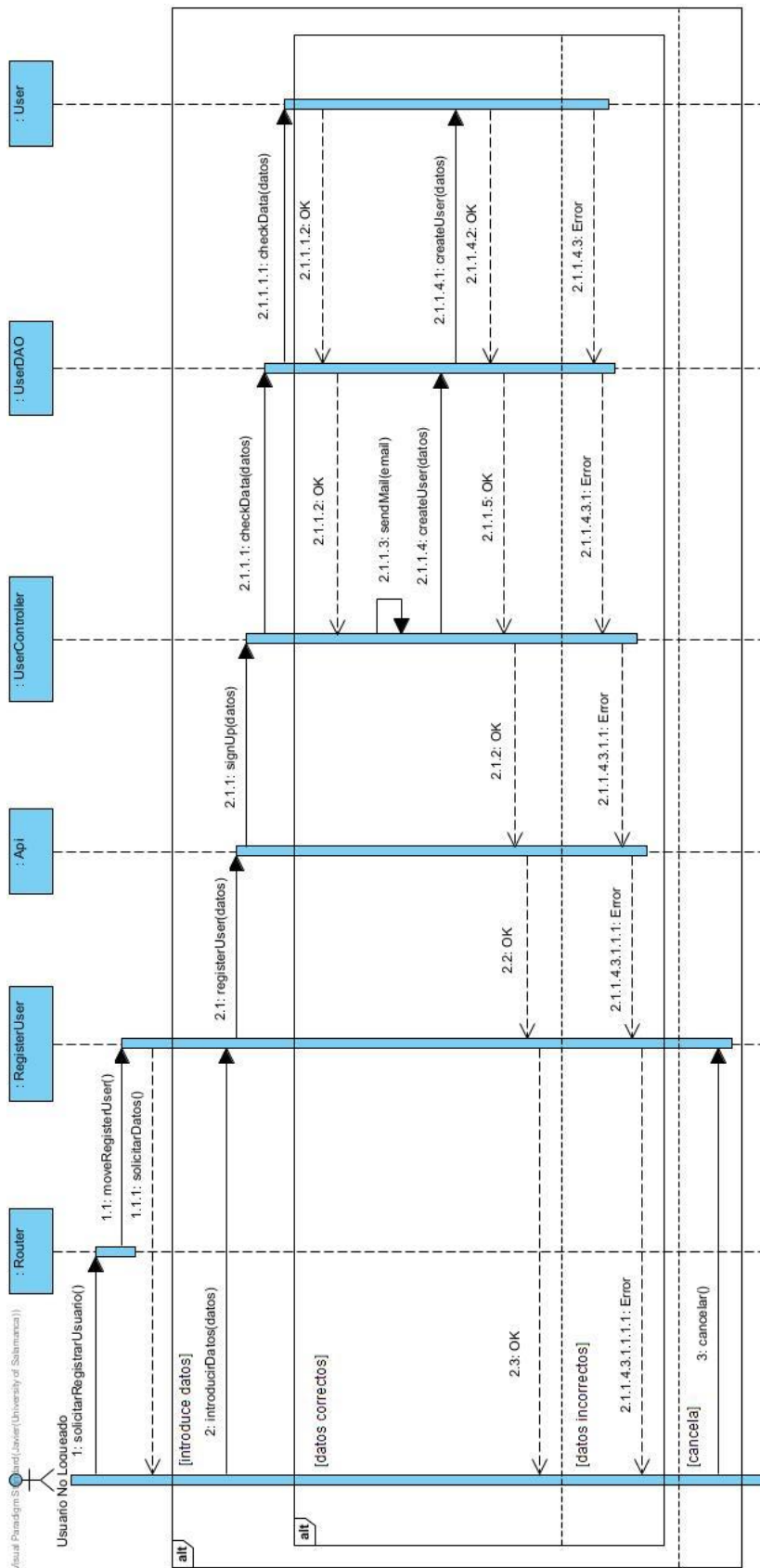


Figura 25: Diagrama de Secuencia Diseño Registrar Usuario

### 6.6.7. Diseño de la Base de Datos

El diseño de la base de datos define la información a almacenar en el sistema junto a sus relaciones. En la siguiente figura se muestra el diseño de la base de datos del sistema.

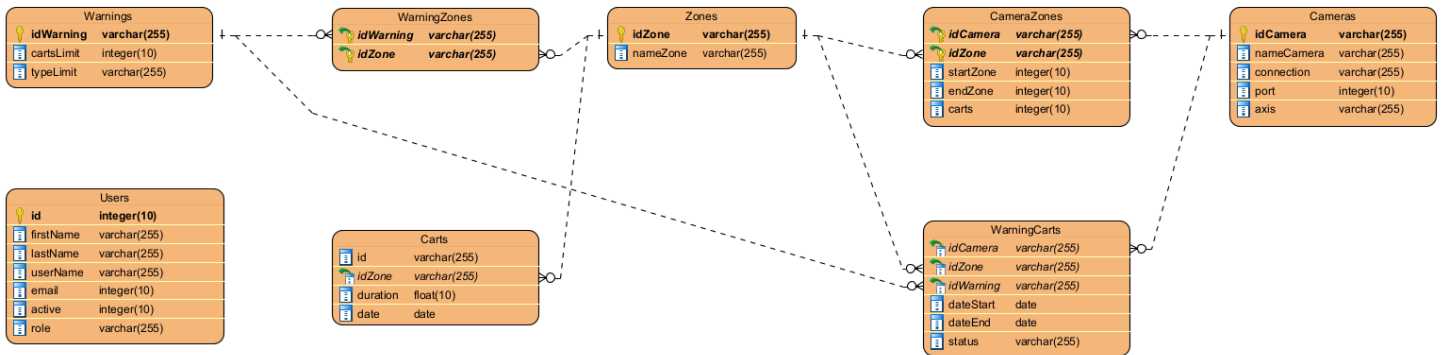


Figura 26: Diseño de la Base de Datos

### 6.6.8. Modelo de Despliegue

El modelo de despliegue permite observar la organización de los nodos del sistema, así como su relación. El diagrama de despliegue del sistema se muestra en la siguiente figura.

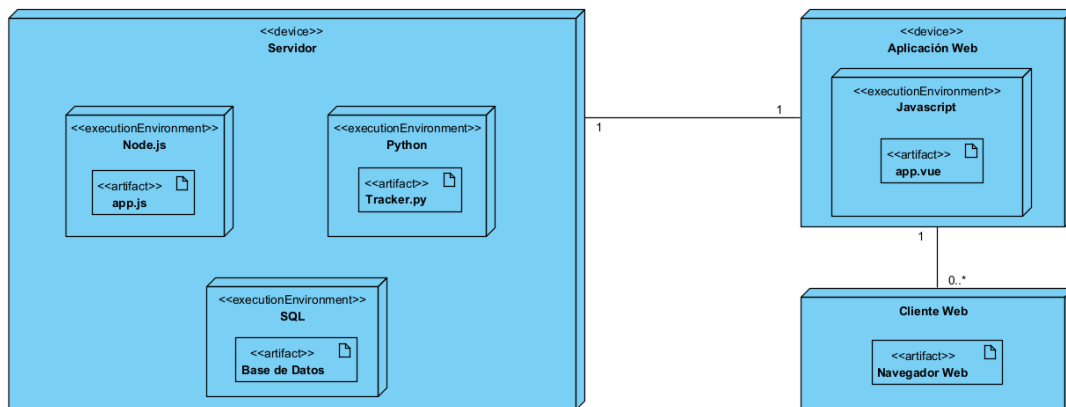


Figura 27: Modelo de Despliegue

El sistema está compuesto por diferentes nodos:

- **Servidor.** El servidor será el encargado de implementar la API para recuperar y almacenar la información del sistema, así como detectar y trackear los carros del supermercado.

- **Aplicación Web.** Es la aplicación que utilizan los usuarios del sistema desarrollada en Vue.js.
- **Cliente Web.** Representa al navegador por el cual el usuario accede a la aplicación web para utilizarla.

## 6.7. Implementación

En la fase de implementación se lleva a cabo la implementación del sistema teniendo como base los resultados obtenidos en la fase de diseño. Para realizar la implementación se utilizan las técnicas y herramientas especificadas en el apartado 5 del documento. La implementación se divide en tres partes:

- **Tracker.** Implementa la funcionalidad de detección y trackeo de los carros. La detección se lleva a cabo utilizando un modelo YOLO v5 entrenado para localizar carros de la compra. El trackeo se realiza calculando la distancia euclídea entre los carros detectados de dos frames consecutivos. Además, se crea un servidor para cada uno de los trackers en ejecución que permite la recuperación de las imágenes de la cámara con los carros localizados y señalados para su visualización en la aplicación web.
- **Aplicación Web.** Codifica la parte del sistema con la que los usuarios interactúan a través de un navegador web. Además, se implementa la funcionalidad que interactúa con el servidor a través de peticiones HTTP para el almacenamiento, recuperación, actualización y eliminación de los datos del sistema.
- **Servidor.** Implementa la API de comunicaciones con la aplicación web. Esta API permite recoger y almacenar, actualizar y eliminar la información del sistema solicitada por la aplicación web.

## 6.8. Pruebas

Las pruebas son una parte importante en el ciclo de vida de un sistema debido a que en ella se comprueba que el sistema funciona de la manera deseada, sin fallos y cumple con los objetivos especificados.

Se realizan pruebas unitarias de cada parte del sistema durante su implementación, así como al completarla. A su vez, se llevan a cabo pruebas completas del sistema para verificar que las diferentes partes se integran de manera correcta.

Como parte importante de las pruebas se encuentra la detección y trackeo de los carros de la compra. Para llevarlas a cabo, han sido necesarios diferentes carros de la compra en un entorno y una cámara desde una perspectiva similar a las cámaras que se instalan en un supermercado.

## 6.9. Funcionalidad del Sistema

En esta sección se muestra el recorrido a través de la aplicación web del sistema propuesto para facilitar y dar a conocer al usuario la misma. Se detallan las funcionalidades más importantes del sistema que puede llevar a cabo el usuario. Debido a que el sistema contempla tres roles y cada uno de ellos tiene diferentes opciones, la sección se divide según estos:

- Usuario No Logueado.
- Usuario.
- Administrador.

Para obtener más información acerca de la funcionalidad del sistema puede consultar el *Anexo VI Manual de Usuario*.

### 6.9.1. Usuario No Logueado

Este usuario realiza las tareas de registro de usuario, activar usuario, iniciar sesión y recuperar contraseña. Tras iniciar sesión se transforma en usuario o administrador dependiendo de su rol dentro del sistema.

#### 6.9.1.1. Iniciar Sesión

El sistema cuando el usuario no logueado abre la aplicación web le muestra la página de iniciar sesión, en ella también se muestran las opciones de registrar usuario o recuperar la contraseña. La imagen del logo de la aplicación de la barra superior se mantiene en las diferentes pantallas previas al ingreso al sistema y permite direccionar al usuario no logueado a la pantalla de inicio de sesión.

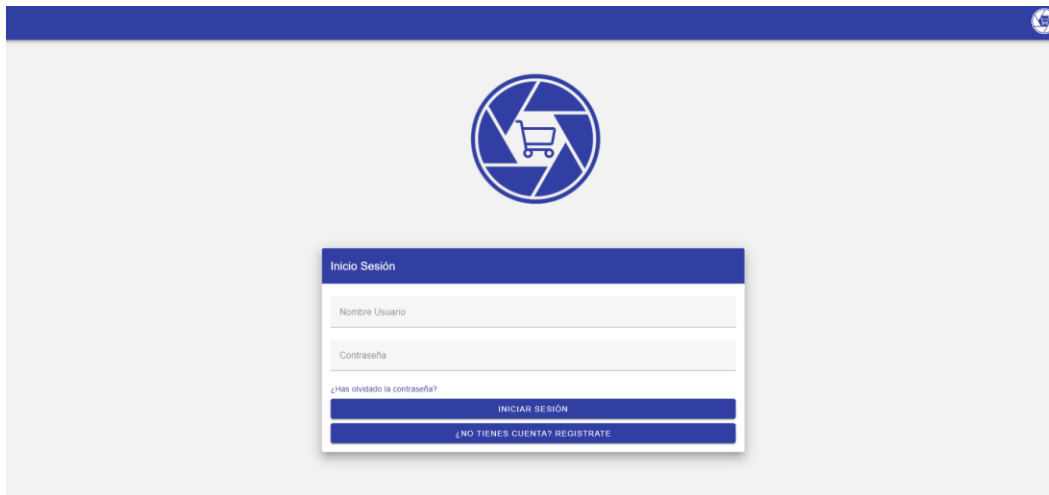


Figura 28: Pantalla Iniciar Sesión

### 6.9.1.2. Registrar Usuario

El usuario si no está registrado en el sistema debe hacerlo para acceder a él. Para ello, debe pulsar el botón *¿No tienes cuenta? Regístrate* en la pantalla de inicio de sesión. En esta opción se muestra un formulario con los datos que debe introducir: nombre, apellidos, nombre de usuario, correo electrónico, contraseña y de nuevo la contraseña para confirmarla. Además, se encuentra el botón *Iniciar Sesión* que devuelve al usuario a la pantalla de inicio de sesión.

Figura 29: Pantalla Registrar Usuario

### 6.9.1.3. Recuperar Contraseña

Para recuperar la contraseña, el usuario pulsa el enlace *¿Has olvidado la contraseña?* de la pantalla de inicio de sesión. Se muestra un formulario en el que es necesario introducir el correo electrónico del usuario.

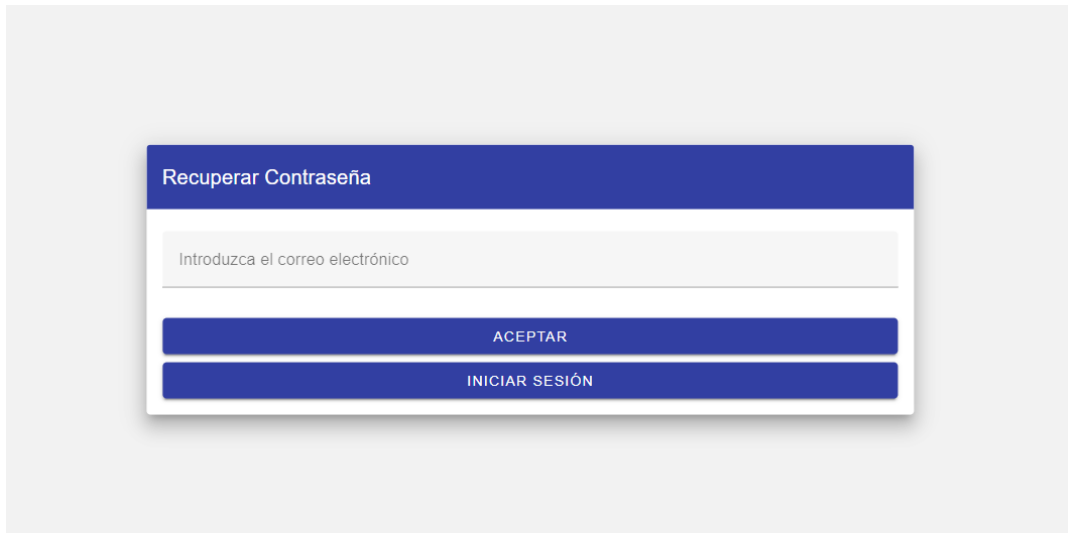
The image shows a web form titled "Recuperar Contraseña" (Recover Password). It features a white input field with the placeholder text "Introduzca el correo electrónico" (Enter the email address). Below the input field are two blue buttons: "ACEPTAR" (Accept) and "INICIAR SESIÓN" (Log In).

Figura 30: Pantalla Correo Electrónico Recuperar Contraseña

En el caso de que se mande el correo electrónico se muestra un mensaje de información.

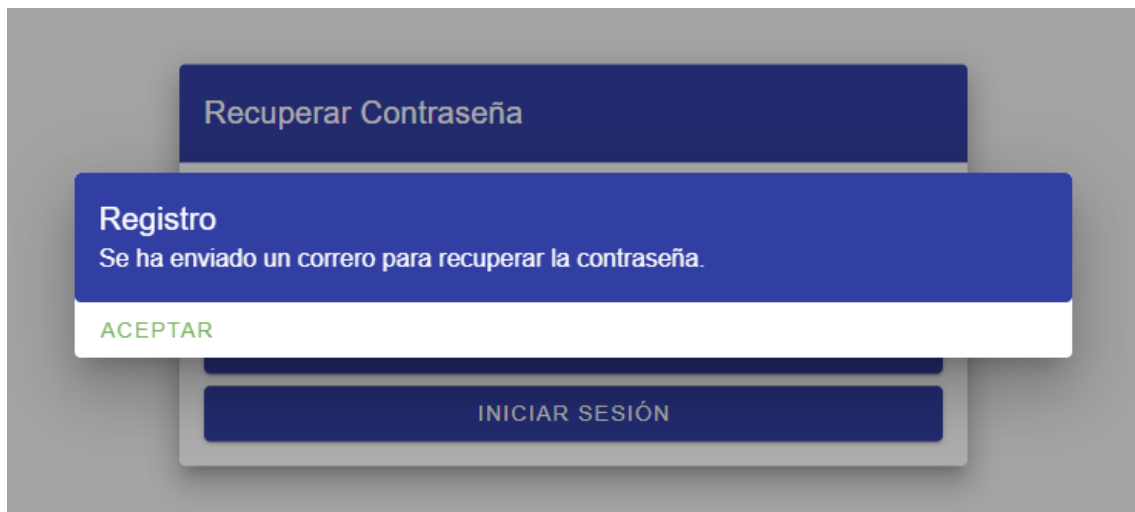
The image shows a confirmation message overlay on the "Recuperar Contraseña" form. The message is titled "Registro" (Registration) and contains the text "Se ha enviado un correo para recuperar la contraseña." (An email has been sent to recover the password.). Below the message is a green "ACEPTAR" (Accept) button. The "INICIAR SESIÓN" (Log In) button from the background form is also visible.

Figura 31: Envío Correo Electrónico Recuperar Contraseña

En el correo electrónico enviado se encuentra un enlace que al pulsarlo redirige a la pantalla para restaurar la contraseña.

# Cambio de contraseña

Hola jose

Pulse el siguiente link para cambiar la contraseña.

[Click aquí](#)

Figura 32: Correo Electrónico Recuperar Contraseña

En esta pantalla se muestra un formulario en el que es necesario introducir la contraseña y su confirmación.



Recuperar Contraseña

Contraseña

Confirmar Contraseña

GUARDAR CONTRASEÑA

Figura 33: Pantalla Introducir Contraseña Recuperar Contraseña

## 6.9.2. Usuario

Todos los usuarios que tienen acceso al sistema pueden visualizar diferentes estadísticas de los carros detectados, los avisos detectados y las cámaras en directo. Además, pueden ver sus datos personales, modificarlos, modificar su contraseña y eliminar permanentemente su cuenta del sistema.

### 6.9.2.1. Inicio

En la pantalla de inicio del sistema se muestran dos gráficos: en el primero se muestran los carros actuales en cada una de las zonas que se actualiza en tiempo real, el segundo muestra un resumen de los últimos 7 días contabilizando el número de carros por día de la semana.

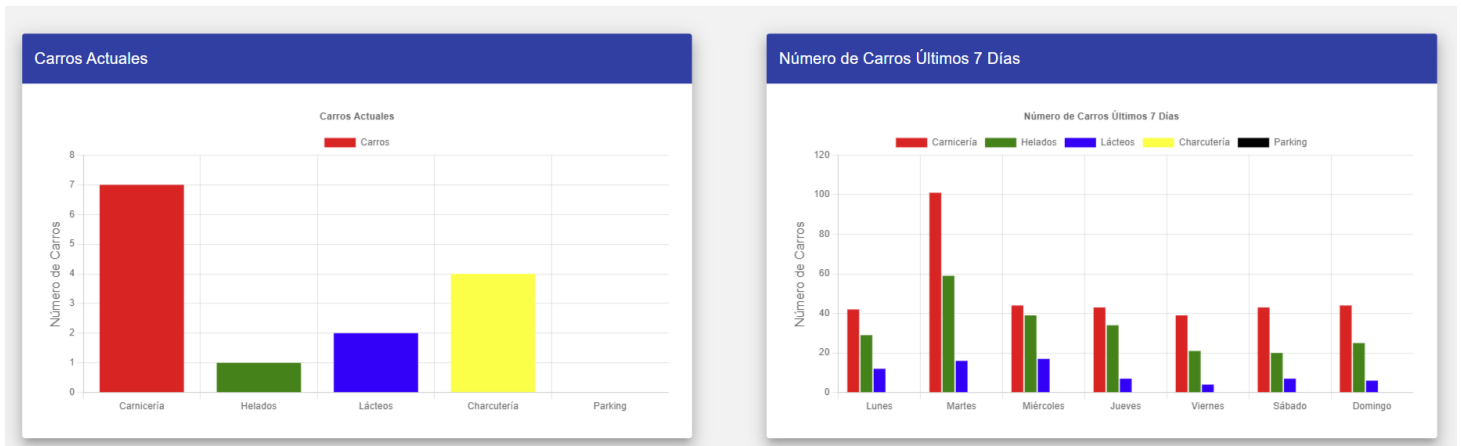


Figura 34: Pantalla Inicio

### 6.9.2.2. Estadísticas Carros

En la parte superior de la pantalla se muestra dos tarjetas, la primera permite seleccionar el tipo de gráficos, barras o líneas. La segunda selecciona el intervalo en el que se realiza la búsqueda de los datos.

**Tipo de Gráficos**

BARRAS

LINEAS

**FECHAS**

Fecha Inicio: 27 Mar 2023

Fecha Fin: 05 Abr 2023

**ACTUALIZAR**

Figura 35: Elección Tipo Gráfico y Fecha

El usuario al pulsar uno de los calendarios, este se despliega y permite al usuario elegir el día deseado. Tras elegir los días, el usuario al pulsar el botón *Actualizar* se el sistema muestra los diferentes gráficos correspondientes a dicho intervalo.

Los gráficos son los mismos tanto para los de barras y de líneas, únicamente varía el tipo de representación. En la siguiente figura se muestra un ejemplo de un gráfico.

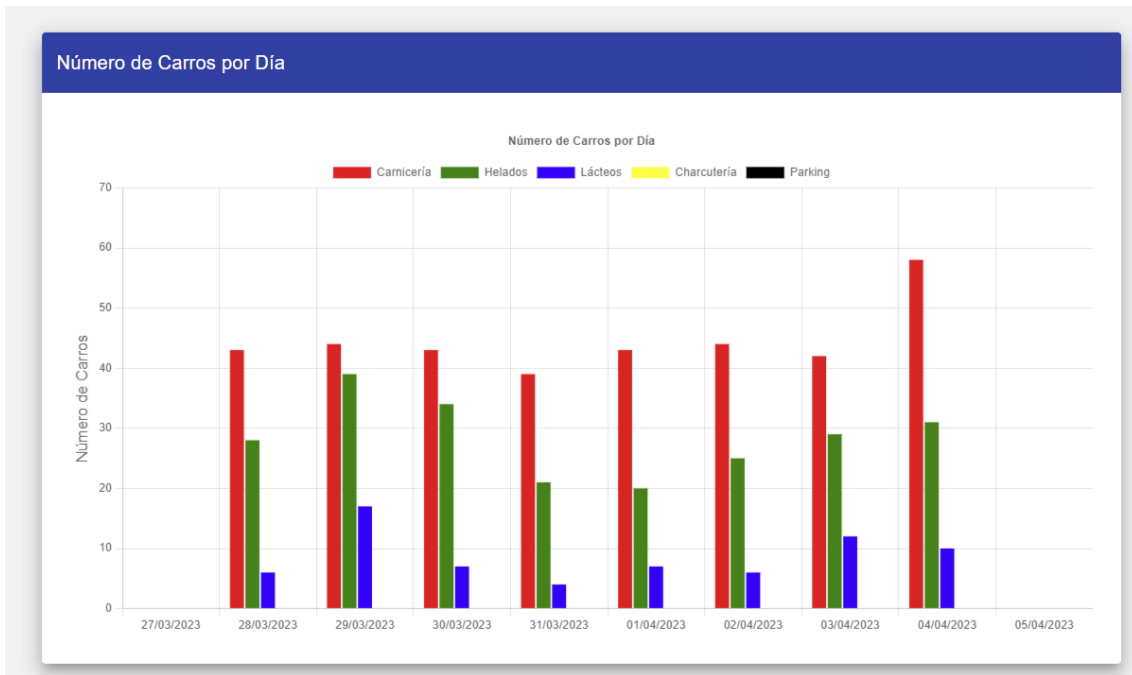


Figura 36: Gráfico Número de Carros por Día

### 6.9.2.3. Estadísticas Avisos

En la parte superior se muestra la misma tarjeta que en la pantalla de estadísticas para elegir el intervalo de fechas. Para elegir los días y actualizar los datos se realiza de la misma manera que en el caso de los gráficos de los carros.

En esta pantalla se muestran los avisos detectados en el intervalo elegido y sus datos. Se observa el identificador del aviso, la zona en la que se ha detectado, la cámara que lo ha detectado, el estado del aviso y la fecha de inicio y fin. En el caso de que el aviso no esté resuelto la fecha de inicio y fin es la misma, tras resolverse el aviso se actualiza la fecha de fin a la fecha en la que se ha resuelto. Además, los campos se pueden ordenar de manera ascendente o descendente pulsando los títulos de las columnas. Por otra parte, se puede buscar a través de la caja de texto que muestra *Buscar*.

ID ↑↓	Zona ↑↓	Cámara ↑↓	Estado ↑↓	Duración ↑↓	Fecha Inicio ↑↓	Fecha Fin ↑↓
1	2	3	Resuelto	16 horas y 58 minutos y 18 segundos	12/05/2023, 19:32:21	13/05/2023, 12:30:39
1	2	3	Resuelto	16 horas y 55 minutos y 43 segundos	12/05/2023, 19:34:56	13/05/2023, 12:30:39
1	2	3	Resuelto	16 horas y 55 minutos y 19 segundos	12/05/2023, 19:35:20	13/05/2023, 12:30:39

« < 1 > » 5 ▾

Figura 37: Pantalla Avisos

#### 6.9.2.4. Cámaras en Directo

En esta pantalla se muestran las imágenes en tiempo real de las cámaras mostrando las zonas de la cámara y los carros detectados. El usuario puede elegir la cámara a visualizar entre las cámaras registradas en el sistema en el selector de la parte izquierda.

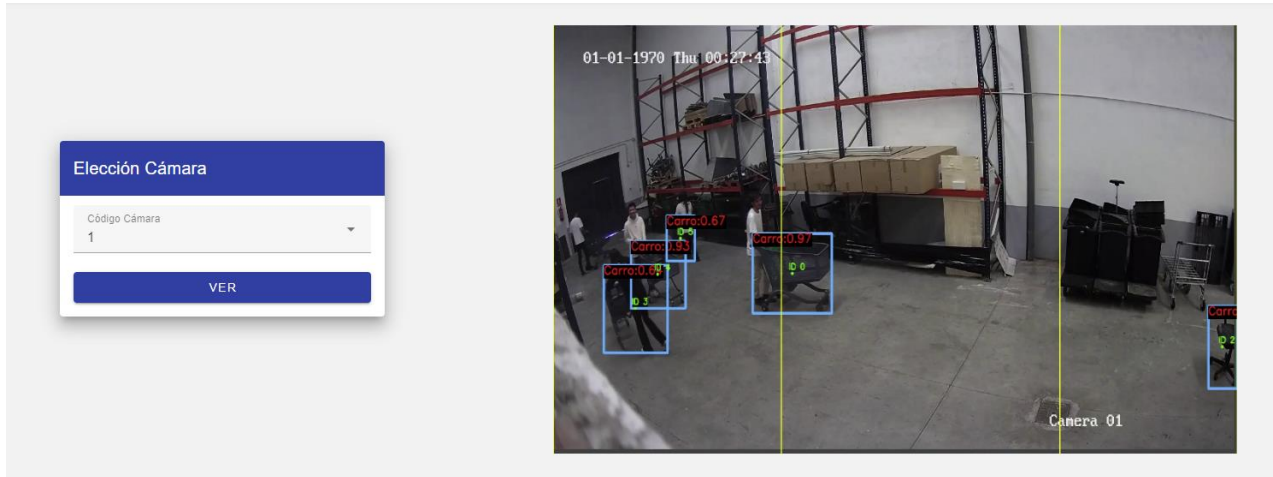


Figura 38: Pantalla Cámaras en Directo

#### 6.9.2.5. Perfil

En esta pantalla se muestran los datos personales del usuario y su rol.



Figura 39: Pantalla Perfil

Al pulsar el botón *Editar*, el sistema muestra un cuadro de diálogo con los campos modificables y su contenido son los datos actuales del usuario. Al pulsar el botón *Editar Perfil* se actualizan los datos del usuario con los datos introducidos, en el caso de pulsar el botón *Cerrar*, los datos no se modifican.




Figura 40: Pantalla Editar Perfil

#### 6.9.2.6. Modificar Contraseña

Se muestra un formulario en el que es necesario introducir la contraseña actual y posteriormente la nueva contraseña. Tras pulsar el botón *Guardar Contraseña* se modifica la contraseña del usuario.



Figura 41: Pantalla Modificar Contraseña

### 6.9.2.7. Eliminar Cuenta

El sistema muestra un mensaje de confirmación que el usuario debe aceptar para eliminar permanentemente su cuenta. Si el usuario pulsa el botón *Aceptar* el usuario es eliminado del sistema y devuelve al usuario a la pantalla de inicio de sesión, en caso de pulsar el botón *Cancelar* se cierra el cuadro diálogo y el usuario no es eliminado.

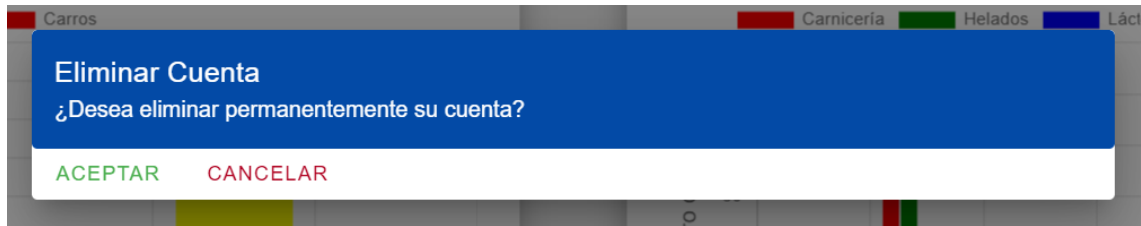


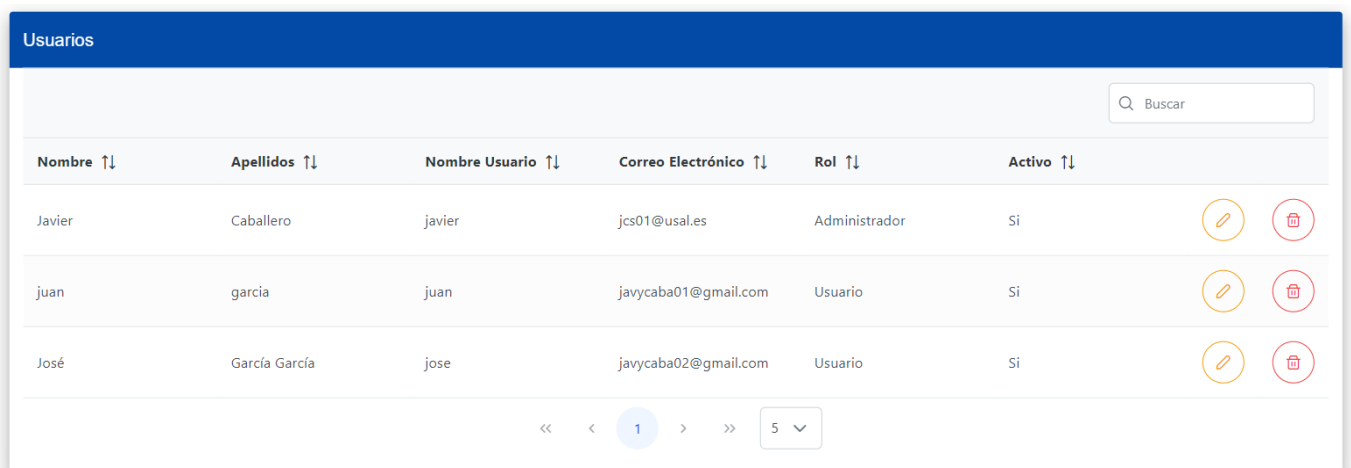
Figura 42: Pantalla Eliminar Cuenta







### 6.9.3. Administrador

El administrador puede realizar todas las tareas que puede el usuario. Además, tiene la capacidad de la gestión de usuarios, zonas, cámaras y avisos.

#### 6.9.3.1. Gestión de Usuarios

El administrador en la pantalla de gestión de usuarios puede visualizar los usuarios registrados en el sistema junto a sus datos.



Nombre ↑↓	Apellidos ↑↓	Nombre Usuario ↑↓	Correo Electrónico ↑↓	Rol ↑↓	Activo ↑↓	
Javier	Caballero	javier	jcs01@usal.es	Administrador	Si	 
Juan	garcía	juan	javycaba01@gmail.com	Usuario	Si	 
José	García García	jose	javycaba02@gmail.com	Usuario	Si	 

Navigation: << < 1 > >> 5 ▾

Figura 43: Pantalla Gestión de Usuarios

En cada fila del usuario se muestran dos botones para la modificación o eliminación de dicho usuario. El botón de modificación está representado por un lápiz y color amarillo mientras que el de eliminación está representado por una papelera y color rojo.

El administrador al pulsar el botón de editar usuario se muestra un cuadro de diálogo en el que se muestra el nombre de usuario a modificar y su rol. Solo es modificable el campo de rol. Al pulsar el botón *Editar Usuario* se actualiza el rol del usuario, en el caso de pulsar el botón *Cancelar*, el rol no se modifica.

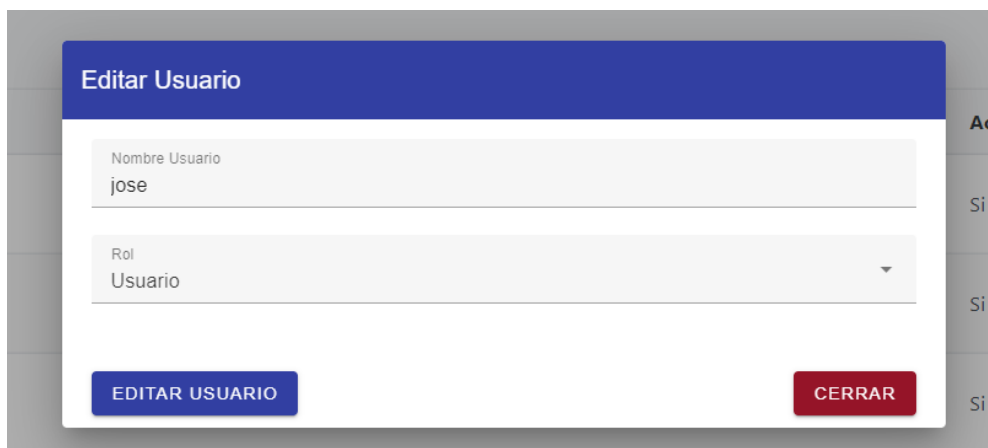


Figura 44: Pantalla Editar Usuario

El administrador al pulsar el botón de eliminar usuario se muestra un cuadro de diálogo en el que se pide la confirmación para eliminar definitivamente al usuario del sistema. Si el administrador pulsa el botón *Aceptar*, el usuario es eliminado del sistema, en el caso de pulsar *Cancelar* el usuario no es eliminado.

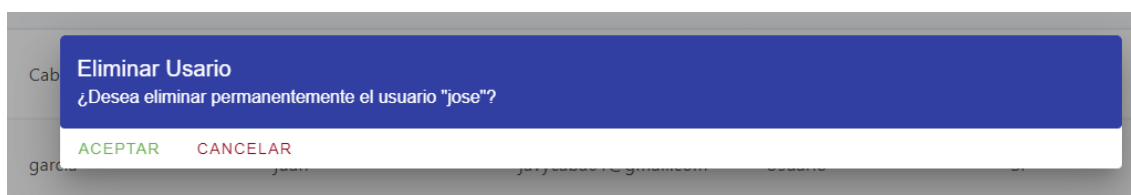


Figura 45: Pantalla Eliminar Usuario

### 6.9.3.2. Gestión de Zonas

El administrador en la pantalla de gestión de zonas puede visualizar las zonas registradas en el sistema junto a sus datos, el identificador y el nombre de la zona.

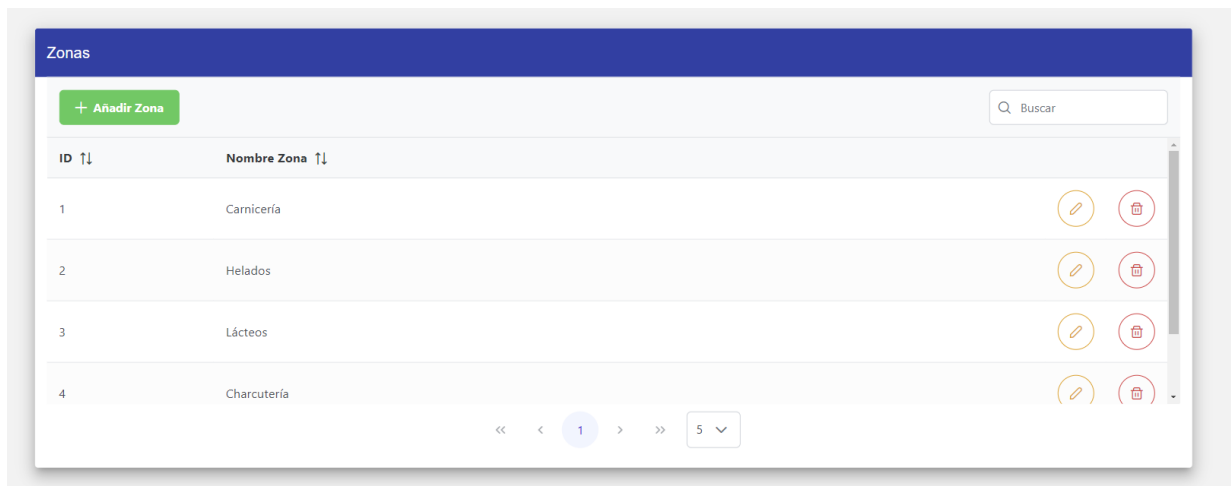


Figura 46: Pantalla Gestión de Zonas

En cada fila de la zona se muestran dos botones, tienen la misma representación que en la tabla de los usuarios, para la modificación o eliminación de dicha zona. Por otra parte, el administrador puede añadir una zona pulsando el botón *Añadir Zona*, situado en la esquina superior izquierda de la tabla.

El sistema muestra un cuadro de diálogo para que el administrador introduzca el id y el nombre de la zona. Al pulsar el botón *Añadir Zona* y los datos son correctos, se añade la nueva zona al sistema. En el caso de pulsar *Cerrar* el cuadro de diálogo se cierra sin realizar ninguna acción.

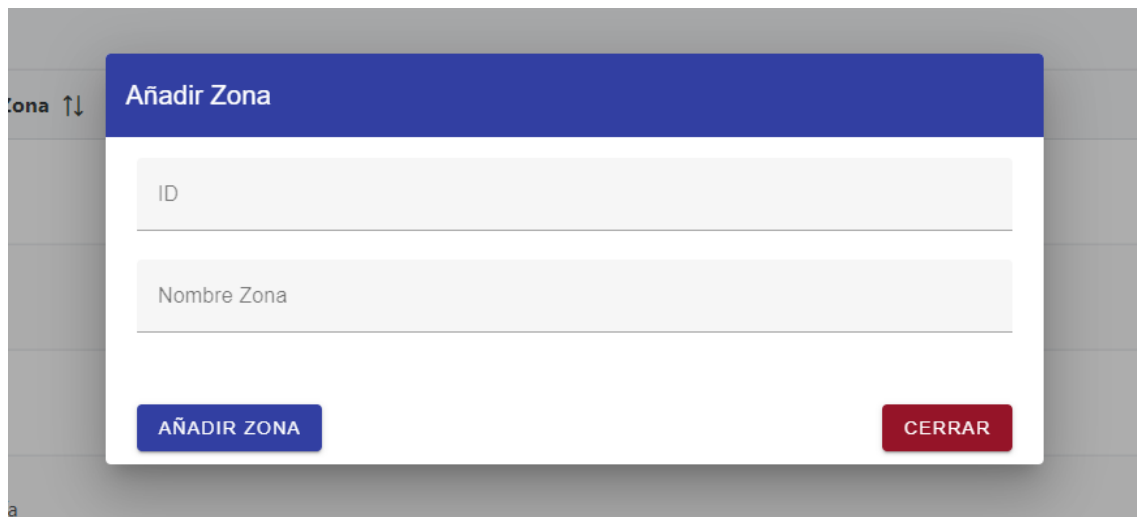


Figura 47: Pantalla Añadir Zona

El administrador al pulsar el botón de editar zona se muestra un cuadro de diálogo en el que se muestra el id y el nombre de la zona. Solo es modificable el campo del nombre. Al pulsar el botón *Editar Zona* y los datos son correctos, se edita el nombre de la zona. En el caso de pulsar *Cerrar* el cuadro de diálogo se cierra sin realizar ninguna acción.

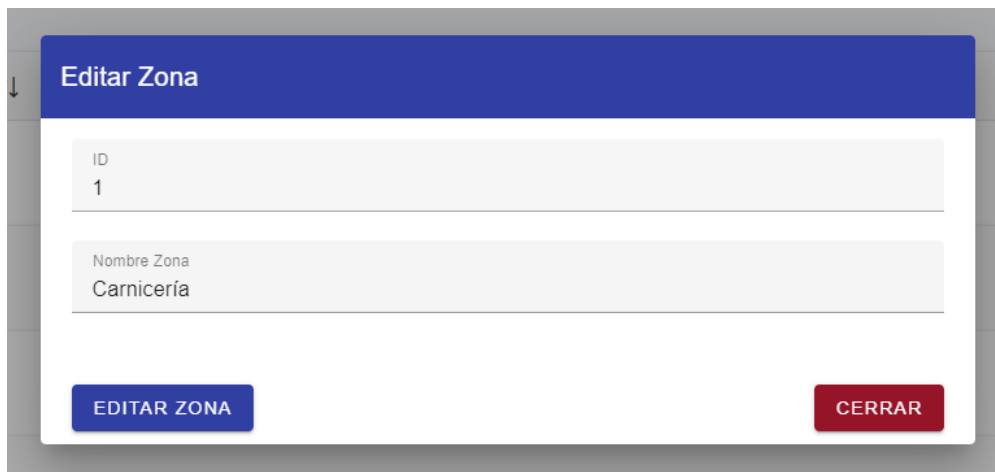


Figura 48: Pantalla Editar Zona

El administrador al pulsar el botón de eliminar zona se muestra un cuadro de diálogo en el que se pide la confirmación para eliminar definitivamente a la zona del sistema. Al pulsar el botón *Aceptar* se elimina la zona del sistema. En el caso de pulsar *Cancelar* el cuadro de diálogo se cierra sin realizar ninguna acción.

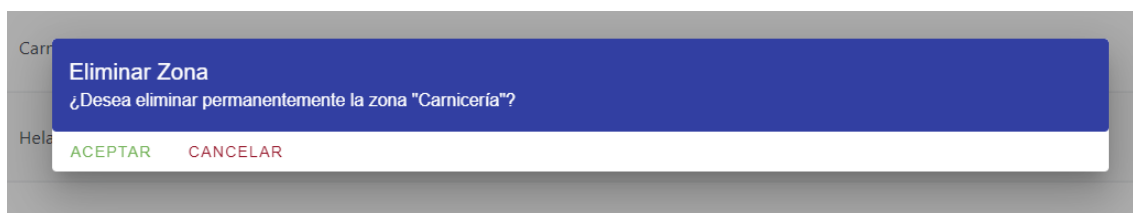


Figura 49: Pantalla Eliminar Zona

### 6.9.3.3. Gestión de Cámaras

El administrador en la pantalla de gestión de cámaras puede visualizar las cámaras registradas en el sistema junto a sus datos, id, conexión, puerto y eje y las zonas que tiene asociadas junto a sus coordenadas.

Cámaras						
+ Añadir Cámara						
ID Cámara ↑↓	Conexión ↑↓	Puerto ↑↓	Eje ↑↓	ID Zona ↑↓	Comienzo Zona	Fin Zona
1	192.168.1.64	5000	X	1	0	640
				2	640	1422
				3	1422	1920
2	192.168.1.65	5001	X	3	0	500
				4	500	1920

Figura 50: Pantalla Gestión de Cámaras

En cada fila de la zona se muestran dos botones, tienen la misma representación que en la tabla de los usuarios, para la modificación o eliminación de dicha cámara. Por otra parte, el administrador puede añadir una cámara pulsando el botón *Añadir Cámara*, situado en la esquina superior izquierda de la tabla.

El sistema muestra un cuadro de diálogo para que el administrador introduzca el id, la conexión, puerto y orientación de las zonas de la cámara. El administrador puede añadir una zona pulsando el botón verde con el símbolo “+”, de la misma manera puede eliminar una zona pulsando el botón rojo con el símbolo “x”. Al pulsar el botón *Añadir Cámara* y los datos son correctos, se añade la nueva cámara al sistema. En el caso de pulsar *Cerrar* el cuadro de diálogo se cierra sin realizar ninguna acción. Además, se debe introducir al menos una zona que controle dicha cámara.

**Añadir Cámara**

ID  
5

---

Conexión  
192.168.1.50

---

Puerto  
5005

---

Eje Zonas  
X

---

Código Zona  
1 x

---

Comienzo Zona  
0

---

Fin Zona  
500

Figura 51: Pantalla Añadir Cámara 1

Eje Zonas  
X

---

Código Zona  
1

Comienzo Zona  
0

Fin Zona  
500

Código Zona  
2

Comienzo Zona  
500

Fin Zona  
1920

+

AÑADIR CÁMARA

CERRAR

Figura 52: Pantalla Añadir Cámara 2

El administrador al pulsar el botón de editar cámara se muestra un cuadro de diálogo en el que se muestra los datos de la cámara. Son modificables todos los campos excepto el identificador de la cámara. Si al pulsar el botón *Editar Cámara* se produce alguno de los errores contemplados en la adicción de una cámara se muestra de la misma manera al administrador.

Editar Cámara

ID  
1

Conexión  
192.168.1.64

Puerto  
5000

Eje Zonas  
X

---

Código Zona  
1

Comienzo Zona  
0

Fin Zona  
640

Código Zona  
2

Figura 53: Pantalla Editar Cámara 1

Fin Zona  
640

Código Zona  
2

Comienzo Zona  
640

Fin Zona  
1422

Código Zona  
3

Comienzo Zona  
1422

Fin Zona  
1920

+

EDITAR CÁMARA

CERRAR

Figura 54: Pantalla Editar Cámara 2

El administrador al pulsar el botón de eliminar cámara se muestra un cuadro de diálogo en el que se pide la confirmación para eliminar definitivamente a la cámara del sistema. Al pulsar el botón Aceptar se elimina la cámara del sistema. En el caso de pulsar Cancelar el cuadro de diálogo se cierra sin realizar ninguna acción.

192

Eliminar Cámara

¿Desea eliminar permanentemente la cámara "1"?

ACEPTAR CANCELAR

Figura 55: Pantalla Eliminar Cámara

#### 6.9.3.4. Gestión de Avisos

El administrador en la pantalla de gestión de avisos puede visualizar los avisos registrados en el sistema junto a sus datos, el id, el límite de carros y el tipo de límite y las zonas que tiene asociadas.

ID Aviso ↑↓	Límite Carros ↑↓	Tipo Límite ↑↓	ID Zona ↑↓
1	1	Inferior	1
2	0	Inferior	3
3	15	Superior	2
			3
			5

Figura 56: Pantalla Gestión de Avisos

En cada fila del aviso se muestran dos botones, tienen la misma representación que en la tabla de los usuarios, para la modificación o eliminación de dicho aviso. Por otra parte, el administrador puede añadir un aviso pulsando el botón *Añadir Aviso*, situado en la esquina superior izquierda de la tabla.

El sistema muestra un cuadro de diálogo para que el administrador introduzca el id, la conexión, puerto y orientación de las zonas de la cámara. El administrador puede añadir una zona pulsando el botón verde con el símbolo “+”, de la misma manera puede eliminar una zona pulsando el botón rojo con el símbolo “x”.

**Añadir Aviso**

---

x

+

AÑADIR AVISO
CERRAR

Figura 57: Pantalla Añadir Aviso

Si al pulsar el botón *Añadir Aviso* no se cumplen los siguientes requisitos se muestra un error indicando el motivo: alguno de los campos no está relleno, el límite de carros debe ser un número entero positivo, se introducen zonas duplicadas o no se introduce ninguna zona.

El administrador al pulsar el botón de editar aviso se muestra un cuadro de diálogo en el que se muestra los datos del aviso. Son modificables todos los campos excepto el identificador del aviso. Además, se pueden añadir o eliminar zonas asociadas.

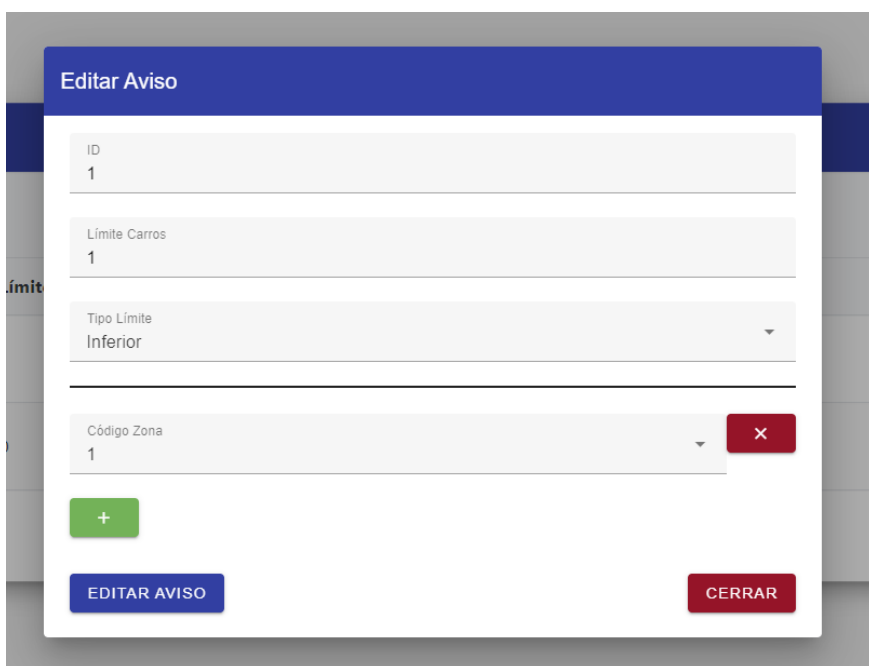


Figura 58: Pantalla Editar Aviso

El administrador al pulsar el botón de eliminar aviso se muestra un cuadro de diálogo en el que se pide la confirmación para eliminar definitivamente al aviso del sistema. Al pulsar el botón *Aceptar* se elimina el aviso del sistema. En el caso de pulsar *Cancelar* el cuadro de diálogo se cierra sin realizar ninguna acción.

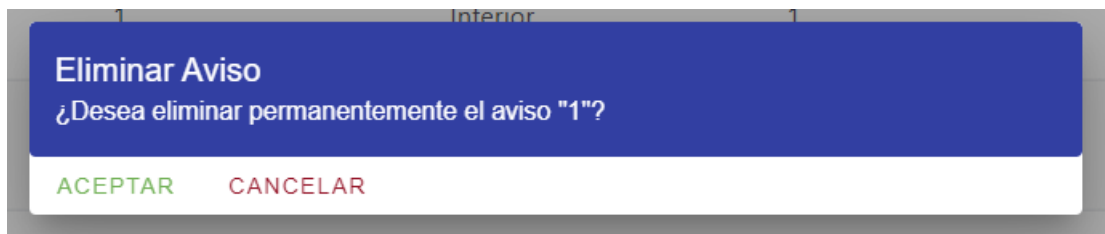


Figura 59: Pantalla Eliminar Aviso

## 7. Resultados Entrenamiento Modelo

Para tener la capacidad de detectar un objeto en una imagen es necesario utilizar un modelo y entrenarlo con un dataset, conjunto de imágenes en el que se etiquetan los objetos a detectar, determinado para los objetos a localizar. Este dataset debe ser suficientemente grande y variado para conseguir resultados correctos.

El conjunto de imágenes que se han utilizado para etiquetar los carros lo componen 283 imágenes en las que se han etiquetado 1006 carros. Para generar un dataset más completo, con la ayuda de la herramienta Roboflow, se ha modificado la saturación, brillo de las imágenes y variado el brillo y la exposición de los cuadrados delimitadores generando un dataset de 677 imágenes. Se han utilizado 591 imágenes para el entrenamiento del modelo, 55 para la validación y 31 para la comprobación del modelo.

Para determinar si una predicción es correcta o no se utiliza el término IoU (Intersection over Union) que mide el solapamiento entre la caja de la predicción y el objeto. Se considera que una predicción es correcta si este valor es mayor a 0.5.

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} \quad [35]$$

Existen diferentes métricas obtenidas como resultado tras el entrenamiento del modelo. Algunas de ellas son:

- **Precisión.** Mide cómo de correctas son las predicciones. Permite conocer el porcentaje de acierto de los objetos a localizar. Si este valor es alto, indica que los objetos localizados son los buscados, pero no tiene en cuenta los objetos a localizar no encontrados.

$$Precision = \frac{TP}{TP + FP} \quad [35]$$

- **Recall.** Esta métrica mide como de bien localiza los objetos que se buscan. Si este valor es alto, indica que el modelo reconoce un gran porcentaje de los objetos a buscar, pero no tiene en cuenta los objetos de predicciones erróneas.

$$Recall = \frac{TP}{TP + FN} \quad [35]$$

TP = Positivo verdadero, TN = Negativo verdadero, FN = Falso negativo.

- **mAp\_0.5 y mAp\_0.5:0.95.** La métrica Average Precision (AP) es el área debajo de la curva precisión-recall.

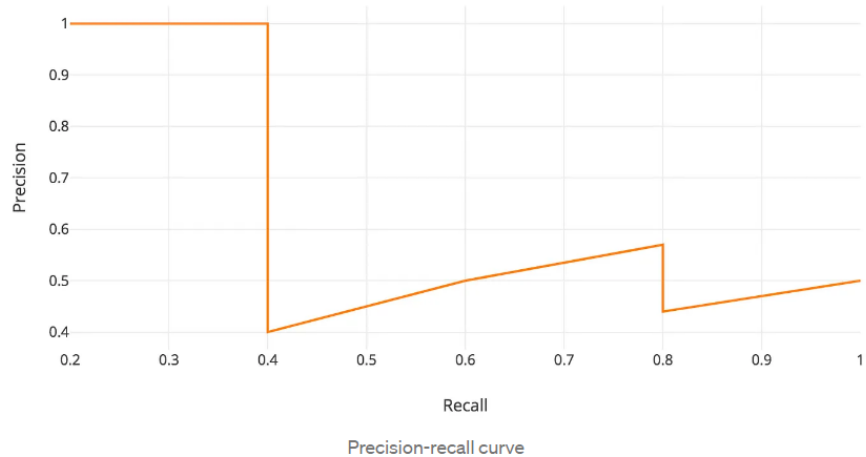


Figura 60: Curva Precisión-Recall [36]

Para calcular la métrica Mean Average Precision (mAP), en el proyecto se utiliza COCO mAP, que usa un rango de valores límite de IoU para cada uno de los objetos y posteriormente realiza la media.

COCO utiliza 101 valores de recall [0:.01:1]. Para la métrica mAp\_0.5:0.95 se utilizan los valores de 0.5 a 0.95 con intervalos de 0.05. El valor obtenido para el valor 0.5 de IoU se denomina mAp\_0.5.

El modelo YOLO v5 utilizado tras entrenarlo 200 epochs con el dataset creado ha obtenido los siguientes resultados:

- Precisión = 0.982.
- Recall = 0.969.
- mAp\_0.5 = 0.983.
- mAp\_0.5:0.95 = 0.697.

## 8. Limitaciones del Sistema

La limitación principal encontrada en el sistema desarrollado se encuentra en la detección y trackeo de los carros de la compra. Para realizar estas tareas es necesario tener grandes recursos de computación debido a que es un proceso continuo y pesado computacionalmente. Además, a medida que aumenta el número de cámaras con las que se localizan los carros es necesario aumentar la capacidad computacional para que el rendimiento y la calidad no sea perjudicada.

Para resolver o reducir dicho problema no se lleva a cabo la detección de los carros, computacionalmente más pesado que el trackeo, todos los frames recogidos por la cámara sino cada 10 frames a costa de perder ligeramente la calidad del trackeo.

## 9. Conclusiones y Líneas de Trabajo Futuras

En esta sección se exponen las conclusiones obtenidas tras realizar todas las fases del proyecto. Por otra parte, se explican posibles líneas de trabajo en el futuro para mejorar el sistema.

### 9.1. Conclusiones

Tras finalizar el sistema se puede comprobar que se han cumplido completar los objetivos y necesidades planteadas al inicio del ciclo de vida del sistema. Los logros conseguidos son:

- Se ha creado un dataset válido, así como entrenado al modelo YOLO v5 que permite desarrollar un sistema capaz de localizar y trackear carros de la compra a través de las imágenes de una cámara.
- La aplicación web implementada permite visualizar los datos, organizados a través de diferentes gráficos, recogidos por el Tracker permitiendo mejorar la calidad del servicio que proporciona un supermercado a sus clientes a través de analizar dichos datos. Además, es posible visualizar en tiempo real las cámaras facilitando la detección de posibles problemas en una zona del supermercado.
- Por otra parte, la aplicación web permite la gestión del perfil de los usuarios y para los usuarios administradores permite gestionar las zonas, cámaras y avisos dando la posibilidad de personalizar el sistema para las necesidades de cada supermercado.
- A lo largo del desarrollo se ha comprobado cuál es el esfuerzo necesario para realizar un proyecto de forma individual.
- Por último, se ha logrado poner en práctica los conocimientos y habilidades adquiridas a lo largo de los cursos del grado, así como ampliar el conocimiento en fases del ciclo de vida de un proyecto como el diseño o la implementación.

### 9.2. Líneas de Trabajo Futuras

Existen diferentes aspectos del sistema a mejorar en el futuro. Respecto a la localización y trackeo de los carros, uno de ellos sería tener la capacidad de trackear un carro cuando cambia de cámara para tener el conocimiento del recorrido completo en el supermercado de cada carro. Otro posible aspecto para mejorar sería la capacidad de localizar otros objetos como cestas de la compra o personas.

Por otra parte, la aplicación web se puede mejorar con el feedback recibido de los supermercados que incorporen el sistema tanto en el aspecto UX como en el funcional.

## 10. Bibliografía

- [1] Mark P. Melegrito, Alvin Sarraga Alon, Sammy V. Militante, Yolanda D. Austria, Myriam J. Polinar, and Maria Concepcion A. Mirabueno. Abandoned-cart-vision: Abandoned cart detection using a deep object detection approach in a shopping parking space. In 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAET), pages 1–5, 2021.
- [2] I. Haritaoglu and M. Flickner. Detection and tracking of shopping groups in stores. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 1, pages I–I, 2001.
- [3] André Sales Mendes, Gabriel Villarrubia, Javier Caridad, Daniel H. De La Iglesia, and Juan F. De Paz. Automatic wireless mapping and tracking system for indoor location. Neurocomputing, 338:372–380, 2019.
- [4] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).
- [5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- [6] Distrito K. <https://www.distritok.com/>
- [7] Wyn Enterprise. <https://wyn.grapecity.com/>
- [8] Rest. <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
- [9] YOLO v5. <https://wandb.ai/onlineinference/YOLO/reports/YOLOv5-Object-Detection-on-Windows-Step-By-Step-Tutorial---VmlldzoxMDQwNzk4>
- [10] Roboflow. <https://docs.roboflow.com/>
- [11] Google Colab. <https://research.google.com/colaboratory/faq.html#:~:text=Colaboratory%2C%20or%20E2%80%9CColab%20for,learning%2C%20data%20analysis%20and%20education.>
- [12] Python. <https://builtin.com/software-engineering-perspectives/python>
- [13] Flask. <https://pythonbasics.org/what-is-flask-python/>
- [14] OpenCV. <https://opencv.org/about/>
- [15] NumPy. <https://numpy.org/doc/stable/user/whatisnumpy.html>
- [16] Dlib. <http://dlib.net/>
- [17] JavaScript. <https://en.wikipedia.org/wiki/JavaScript>
- [18] HTML. <https://es.wikipedia.org/wiki/HTML>
- [19] CSS. <https://en.wikipedia.org/wiki/CSS>
- [20] Vue.js. <https://vuejs.org/>
- [21] Vuetify. <https://vuetifyjs.com/>

- [22] PrimeVue. <https://primevue.org/>
- [23] Axios. <https://axios-http.com/>
- [24] Vue-chartjs. <https://vue-chartjs.org/>
- [25] NodeJS. <https://nodejs.org/en/about>
- [26] NPM. <https://www.uxpin.com/studio/blog/what-is-npm/>
- [27] Express. <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js>
- [28] MySQL. <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- [29] Sequelize. <https://medium.com/the-javascript-doj/introduction-to-sequelize-1cbfc2d2d1bf>
- [30] Visual Studio Code. <https://code.visualstudio.com/docs/editor/whyvscode>
- [31] Microsoft Project. <https://www.projectmanager.com/blog/what-is-microsoft-project>
- [32] Moreno García M. N. Transparencias de Gestión de Proyectos, Práctica 2 – Planificación temporal.
- [33] Patrón Modelo-Vista-VistaModelo. <https://www.techtarget.com/whatis/definition/Model-View-ViewMode>
- [34] Patrón DAO. <https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/>
- [35] Detección de Objetos. <https://kharshit.github.io/blog/2019/09/20/evaluation-metrics-for-object-detection-and-segmentation>
- [36] Detección de Objetos. <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>