



 

**TFG del Grado en Ingeniería Informática**

**Control de acceso a edificios de vehículos mediante visión artificial**

**Documentación Técnica**



Presentado por Pablo Sainz Pino

en Universidad de Burgos — 07 de junio de 2025

Tutores: José Manuel Galán Ordax y

José Ignacio Santos Martín

# Índice general

[Índice general 3](#_Toc200231882)

[Índice de figuras 5](#_Toc200231883)

[Índice de tablas 7](#_Toc200231884)

[Plan de Proyecto Software 1](#_Toc200231885)

[A.1. Introducción 1](#_Toc200231886)

[A.2. Planificación temporal 1](#_Toc200231887)

[A.3. Estudio de viabilidad 12](#_Toc200231888)

[Especificación de Requisitos 16](#_Toc200231889)

[B.1. Introducción 16](#_Toc200231890)

[B.2. Objetivos generales 16](#_Toc200231891)

[B.3. Catálogo de requisitos 17](#_Toc200231892)

[B.4. Especificación de requisitos 20](#_Toc200231893)

[Especificación de diseño 27](#_Toc200231894)

[C.1. Introducción 27](#_Toc200231895)

[C.2. Diseño de datos 27](#_Toc200231896)

[C.3. Diseño procedimental 29](#_Toc200231897)

[C.4. Diseño arquitectónico 33](#_Toc200231898)

[C.5. Diseño de interfaces 35](#_Toc200231899)

[Documentación técnica de programación 38](#_Toc200231900)

[D.1. Introducción 38](#_Toc200231901)

[D.2. Estructura de directorios 38](#_Toc200231902)

[D.3. Manual del programador 39](#_Toc200231903)

[D.4. Compilación, instalación y ejecución del proyecto 41](#_Toc200231904)

[D.5. Pruebas del sistema 43](#_Toc200231905)

[Documentación de usuario 46](#_Toc200231906)

[E.1. Introducción 46](#_Toc200231907)

[E.2. Requisitos de usuarios 46](#_Toc200231908)

[E.3. Instalación 46](#_Toc200231909)

[E.4. Manual del usuario 47](#_Toc200231910)

[Anexo de sostenibilización curricular 63](#_Toc200231911)

[F.1. Introducción 63](#_Toc200231912)

[F.2. Competencias en sostenibilidad adquiridas 63](#_Toc200231913)

[F.3. Impacto social y ambiental 64](#_Toc200231914)

[F.4. Conclusión 64](#_Toc200231915)

[Bibliografía 66](#_Toc200231916)

# Índice de figuras

[Figura A.1: Issues Sprint 1 2](#_Toc199721107)

[Figura A.2: Burndown Sprint 1 2](#_Toc199721108)

[Figura A.3: Issues Sprint 2 3](#_Toc199721109)

[Figura A.4: Burndown Sprint 2 3](#_Toc199721110)

[Figura A.5: Issues Sprint 3 4](#_Toc199721111)

[Figura A.6: Burndown Sprint 3 4](#_Toc199721112)

[Figura A.7: Issues Sprint 4 5](#_Toc199721113)

[Figura A.8: Burndown Sprint 4 5](#_Toc199721114)

[Figura A.9: Issues Sprint 5 6](#_Toc199721115)

[Figura A.10: Burndown Sprint 5 6](#_Toc199721116)

[Figura A.11: Issues Sprint 6 8](#_Toc199721117)

[Figura A.12: Burndown Sprint 6 8](#_Toc199721118)

[Figura A.13: Issues Sprint 7 9](#_Toc199721119)

[Figura A.14: Burndown Sprint 7 10](#_Toc199721120)

[Figura A.15: Issues Sprint 8 11](#_Toc199721121)

[Figura A.16: Burndown Sprint 8 11](#_Toc199721122)

[Figura C.1: Estructura tabla vehículos 28](#_Toc199721123)

[Figura C.2: Estructura tabla usuarios 28](#_Toc199721124)

[Figura C.3: Diagrama de secuencia de login 29](#_Toc199721125)

[Figura C.4: Diagrama de secuencia de subida y procesamiento 30](#_Toc199721126)

[Figura C.5: Diagrama de secuencia de visualización de resultados 31](#_Toc199721127)

[Figura C.6: Diagrama de secuencia de registro de usuarios 32](#_Toc199721128)

[Figura C.7: Patrón MVC [2] 34](#_Toc199721129)

[Figura C.8: Patrón Fachada [3] 34](#_Toc199721130)

[Figura C.9: Prototipo de interfaz 35](#_Toc199721131)

[Figura C.10: Interfaz final 36](#_Toc199721132)

[Figura D.1: Conexiones correctas 42](#_Toc199721133)

[Figura D.2: Ejemplo bloque try-except 43](#_Toc199721134)

[Figura E.1: Página de inicio 47](#_Toc199721135)

[Figura E.2: Acceso denegado 47](#_Toc199721136)

[Figura E.3: Página de registro 48](#_Toc199721137)

[Figura E.4: Usuario registrado correctamente 48](#_Toc199721138)

[Figura E.5: Ventana usuario con rol experto 49](#_Toc199721139)

[Figura E.6: Selección de modelo 49](#_Toc199721140)

[Figura E.7: GPU activada 50](#_Toc199721141)

[Figura E.8: GPU desactivada 50](#_Toc199721142)

[Figura E.9: Ventana caso 1 51](#_Toc199721143)

[Figura E.10: Línea dibujada 52](#_Toc199721144)

[Figura E.11: Coordenadas detectadas 53](#_Toc199721145)

[Figura E.12: Vídeo sin procesarse 53](#_Toc199721146)

[Figura E.13: Vídeo procesándose 54](#_Toc199721147)

[Figura E.14: Botón finalizar procesamiento 55](#_Toc199721148)

[Figura E.15: Procesamiento finalizado 55](#_Toc199721149)

[Figura E.16: Vídeo con metadatos fuera de rango 55](#_Toc199721150)

[Figura E.17: Vídeo sin metadatos 55](#_Toc199721151)

[Figura E.18: Formulario de facultades 56](#_Toc199721152)

[Figura E.19: Listado de facultades 56](#_Toc199721153)

[Figura E.20: Coordenadas vídeo fuera de rango 56](#_Toc199721154)

[Figura E.21: Selector de fechas 57](#_Toc199721155)

[Figura E.22: Gráfico de barras 58](#_Toc199721156)

[Figura E.23: Resumen general 59](#_Toc199721157)

[Figura E.24 : Porcentajes por tipo 59](#_Toc199721158)

[Figura E.25 : Estadísticas técnicas 60](#_Toc199721159)

[Figura E.26 : Tabla de facultades 60](#_Toc199721160)

[Figura E.27 : Botón Exportar CSV 61](#_Toc199721161)

[Figura E.28 : Botón Cerrar sesión 61](#_Toc199721162)

[Figura E.29 : Botón cambiar idioma 61](#_Toc199721163)

# Índice de tablas

[Tabla A.1: Costes totales 13](#_Toc199721273)

[Tabla A.2: Tabla de licencias 14](#_Toc199721274)

[Tabla B.1: CU01 – Login de usuarios 20](#_Toc199721275)

[Tabla B.2: CU02 – Subida y configuración de vídeos 20](#_Toc199721276)

[Tabla B.3: CU03 – Detección y conteo de vehículos 21](#_Toc199721277)

[Tabla B.4: CU04 – Identificación de facultad 21](#_Toc199721278)

[Tabla B.5: CU05 – Registro de logs en base de datos 22](#_Toc199721279)

[Tabla B.6: CU06 – Visualización de resultados 22](#_Toc199721280)

[Tabla B.7: CU07 – Exportación de resultados 23](#_Toc199721281)

[Tabla B.8: CU08 – Internacionalización 23](#_Toc199721282)

[Tabla B.9: CU09 – Logout de usuarios 24](#_Toc199721283)

[Tabla B.10: CU10 – Registro de usuarios 24](#_Toc199721284)

*Apéndice A*

# Plan de Proyecto Software

## Introducción

Este primer apartado se dedica a explicar el desarrollo del proyecto, incluyendo su planificación temporal y la viabilidad del proyecto, dividida a su vez en viabilidad económica y viabilidad legal.

## Planificación temporal

Desde el inicio del proyecto, se optó por seguir una metodología ágil, realizando sprints de una o dos semanas. Al final de cada sprint se ha llevado a cabo una reunión para revisar las tareas realizadas y planificar los objetivos del siguiente.

La principal herramienta de gestión ha sido Zube, integrada con el repositorio de Github, lo que ha permitido visualizar de forma clara los issues, registrar el avance del proyecto y asignar story points a cada tarea.

*Apéndice A. Planificación temporal*

**Sprint 1 – 17/03/2025 – 01/04/2025**

Durante las dos primeras semanas de desarrollo se configuró el repositorio del proyecto y se investigó sobre modelos de detección. Asimismo, se eligió el entorno de desarrollo y el editor de texto para la documentación.

Una vez completada la investigación, se implementó un primer prototipo funcional, capaz de realizar la detección de vehículos. También se diseñó una interfaz gráfica básica que permite al usuario cargar vídeos y mostrar los resultados obtenidos.

Tabla

El contenido generado por IA puede ser incorrecto.

Figura A.1: Issues Sprint 1

Gráfico

El contenido generado por IA puede ser incorrecto.

Figura A.2: Burndown Sprint 1

*Apéndice A. Planificación temporal*

**Sprint 2 – 01/04/2025 – 08/04/2025**

El objetivo principal de este sprint fue integrar el modelo de detección dentro de un servidor web usando Flask. Para ello, se investigó sobre el funcionamiento de estos servidores y se implementó una primera versión que incluía una serie de botones que permitía procesar vídeos desde el navegador.

Además, se estudió cómo mejorar el rendimiento de YOLO, ya que en vídeos de alta resolución el procesamiento era algo lento.

Tabla

El contenido generado por IA puede ser incorrecto.

Figura A.3: Issues Sprint 2

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A.4: Burndown Sprint 2

*Apéndice A. Planificación temporal*

**Sprint 3 – 10/04/2025 – 21/04/2025**

En este sprint se abordó la integración de la base de datos con la aplicación. Se investigó sobre las diferentes formas de conexión entre Flask y bases de datos, se seleccionó la más adecuada y se implementó la base de datos inicial que permitiría almacenar los logs de las detecciones de vehículos.

A su vez, se comenzó con la redacción de la memoria y los anexos del proyecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Figura A.5: Issues Sprint 3

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A.6: Burndown Sprint 3

*Apéndice A. Planificación temporal*

**Sprint 4 – 22/04/2025 – 29/04/2025**

Durante esta semana, el objetivo principal fue probar el sistema de detección y crear una tabla donde guardar la información de los vehículos detectados. Para ello, se implementó una tabla de prueba y se desarrolló la funcionalidad de conteo automático de vehículos.

Además, se continuó con la redacción de la memoria y los anexos.

Tabla

El contenido generado por IA puede ser incorrecto.

Figura A.7: Issues Sprint 4

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A.8: Burndown Sprint 4

*Apéndice A. Planificación temporal*

**Sprint 5 – 29/04/2025 – 05/05/2025**

En este sprint se implementó la pantalla de inicio de sesión, incorporando el redireccionamiento según el tipo de usuario, uno básico y otro experto. También, se añadió la funcionalidad de cierre de sesión y se mejoró el diseño de la interfaz utilizando Bootstrap.

Por otra parte, se añadieron los logs en la tabla de prueba, permitiendo registrar las detecciones de forma visual. Además, se habilitó un botón para generar un informe básico y poder exportarlo a un archivo en formato CSV. Se continuó con el desarrollo de los anexos y la memoria.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura A.9: Issues Sprint 5

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A.10: Burndown Sprint 5

*Apéndice A. Planificación temporal*

**Sprint 6 – 05/05/2025 – 12/05/2025**

Este sprint marcó un punto de inflexión en el desarrollo del proyecto, ya que se abordaron tareas que permitieron convertir el proyecto en una aplicación plenamente funcional y adaptable al usuario. Para ello, se implementaron funcionalidades orientadas automatizar y personalizar la detección.

Por una parte, se integraron los metadatos extraídos de los vídeos, permitiendo detectar de forma automática si el vídeo pertenece a una de las facultades de la Universidad de Burgos. También, se contempló el caso de que ciertos vídeos pueden no contener metadatos, por lo que en este caso se muestra un formulario para que el usuario seleccione a qué facultad pertenece el vídeo subido. Todos estos datos se almacenan en la base de datos.

Por otra parte, se introdujo la opción de definir una línea de conteo personalizada, que el usuario puede dibujar manualmente sobre el primer frame del vídeo. Esta línea queda registrada, y se realiza el conteo a partir de los vehículos que la cruzan.

Además, se arreglaron varios errores que afectaban al flujo que debía seguir la aplicación, así como fallos visuales que se producían al interactuar con diferentes elementos de la interfaz. Se avanzó también con la documentación.

*Apéndice A. Planificación temporal*

Tabla

El contenido generado por IA puede ser incorrecto.

Figura A.11: Issues Sprint 6

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A.12: Burndown Sprint 6

*Apéndice A. Planificación temporal*

**Sprint 7 – 12/05/2025 – 24/05/2025**

En este sprint se implementaron funcionalidades centradas en la explotación y visualización de datos.

Se desarrolló un gráfico de barras con paginación, además de añadir nuevos campos en los informes. A su vez, se actualizaron los campos de los informes CSV, ya que estaban desfasados, y se mejoró algún aspecto de la interfaz.

Por otro lado, se implementó una tabla con, que permite al usuario consultar y analizar los datos registrados, También se incorporaron ciertas estadísticas clave de los resultados obtenidos

Finalmente, se avanzó en la redacción de memoria y anexos.

Aplicación

El contenido generado por IA puede ser incorrecto.

Figura A.13: Issues Sprint 7

*Apéndice A. Planificación temporal*

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A.14: Burndown Sprint 7

**Sprint 8 – 24/05/2025 – 06/06/2025**

En este sprint se realizó la internacionalización de la aplicación. Se crearon dos diccionarios, uno en español y otro en inglés, que contienen todos los textos de la aplicación para poder cambiar entre idiomas.

A su vez, se implementó el sistema de registro de usuarios, para que el usuario cree un perfil con las credenciales que desee, y se almacenen los datos en la base de datos.

Para finalizar, se realizó el análisis de código con SonarCloud para ver que era posible mejorar. Se realizaron mejoras en temas de seguridad y se consiguió que todo el repositorio pasase la prueba con buena nota.

Se terminó de redactar la documentación y se dieron los detalles finales.

*Apéndice A. Planificación temporal*

Tabla

El contenido generado por IA puede ser incorrecto.

Figura A.15: Issues Sprint 8

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Figura A.16: Burndown Sprint 8

*Apéndice A. Estudio de viabilidad*

## Estudio de viabilidad

En esta sección se van a calcular los costes del proyecto.

**Viabilidad económica**

En este apartado se estiman los costes derivados del desarrollo del proyecto. Por ello, se incluye el coste personal, los recursos materiales y servicios utilizados.

* **Costes de personal**

En este apartado, se va a calcular el gasto que supondría tener al empleado contratado. Se estima que se han dedicado unas 400 horas repartidas a lo largo de 3 meses, lo que supone una media de entre 25 y 35 horas semanales. Se van a estimar un valor medio de 30 horas a la semana. El salario del alumno será de 15€/hora. Esto nos da un salario bruto mensual de:

Este cálculo corresponde al salario bruto mensual del empleado. Sin embargo, para estimar el coste real que tendría para la empresa, es necesario tener en cuenta las cotizaciones dentro del Régimen General de la Seguridad Social [1]:

* 23,60% contingencias comunes
* 5,50% desempleo
* 0,20% FOGASA
* 0,60% formación profesional
* 0,67% Mecanismo de Equidad Intergeneracional (MEI)

*Apéndice A. Estudio de viabilidad*

Todo esto sumado implica un coste adicional del 30,57% para la empresa. Aplicando este porcentaje al salario mensual estimado, el gasto que supone el empleado es de:

* **Costes de hardware**

El proyecto se ha desarrollado íntegramente en un ordenador personal, cuyo coste fue de 1300€. Dado que fue adquirido hace cuatro años, se considera completamente amortizado. Por lo tanto, los costes de hardware se estiman en 0€.

* **Costes de software**

Durante el desarrollo del proyecto, todas las herramientas software utilizadas han sido gratuitas. No ha sido necesaria la adquisición de ninguna licencia ni software de pago, luego el coste total de software ha sido de 0€.

* Total

Teniendo en cuenta los elementos anteriores, el coste total estimado del proyecto asciende a:

|  |  |
| --- | --- |
| **Tipos de costes** | **Total** |
| *Personal* | 7777,61€ |
| *Hardware* | 0€ |
| *Software* | 0€ |
| **Total** | 7777,61€ |

Tabla A.1: Costes totales

*Apéndice A. Estudio de viabilidad*

**Viabilidad legal**

En este apartado se van a mostrar las licencias de cada herramienta utilizada.

|  |  |  |
| --- | --- | --- |
| **Herramienta/Librería** | **Versión** | **Licencia** |
| Python | 3.11.1 | PSF |
| Github | 3.4.18 | MIT |
| Visual Studio Code | 1.100.2 | MIT |
| Flask | 3.1.0 | BSD-3-Clause |
| torch | 2.5.1+cu121 | BSD |
| ultralytics | 8.3.99 | AGPL-3.0 |
| opencv-python | 4.11.0.86 | Apache 2.0 |
| pymediainfo | 7.0.1 | MIT |
| shapely | 2.1.0 | BSD-3-Clause |
| MySQL | 8.0.42 | GNU GPLv2 |
| mysql-connector-python | 9.3.0 | GNU GPLv2 |
| Chart.js | 4.4.9 | MIT |
| Bootstrap | 5.3.0 | MIT |
| DataTables | 1.13.4 | MIT |

Tabla A.2: Tabla de licencias

Como se puede observar, la mayoría de las librerías cuentan con licencias permisivas como MIT, BSD o Apache 2.0.

Sin embargo, ultralytics se encuentra bajo una licencia AGPL-3.0, algo más restrictiva que GPL. Esto implica que cualquier software derivado de esta librería debe heredar su licencia, luego la licencia más adecuada para el proyecto sería *AGPL-3.0*.

*Apéndice B*

# Especificación de Requisitos

## Introducción

En esta sección se enumerarán los requisitos y objetivos con los que contará el proyecto.

## Objetivos generales

El proyecto busca alcanzar los siguientes objetivos:

* Detectar y contar los vehículos que acceden a los campus de la UBU.
* Almacenar los datos obtenidos.
* Contribuir a la monitorización de la movilidad en las facultades.
* Generar informes y representaciones visuales a partir de los datos obtenidos.
* Facilitar la visualización de los datos obtenidos mediante una interfaz web.
* Generar estadísticas a partir de datos obtenidos.

*Apéndice B. Especificación de requisitos*

## Catálogo de requisitos

En este apartado se definen los requisitos funcionales y no funcionales del proyecto.

**Requisitos funcionales**

* **RF1–Login de usuarios:** La aplicación debe permitir al usuario iniciar sesión mediante credenciales válidas.
* **RF2-Carga de vídeos**: La aplicación debe permitir al usuario subir un vídeo para su procesamiento, seleccionar el modelo de YOLOv11 a utilizar y configurar si el procesamiento lo quiere realizar con CPU o GPU, siempre y cuando sea esto último sea posible. También debe permitir finalizar el procesamiento.
* **RF3-Detección de vehículos:** La aplicación debe ser capaz de detectar y contar vehículos automáticamente mediante un modelo de visión por computador.
  + **RF3.1-Uso de YOLOv11**: El sistema debe utilizar el modelo YOLOv11 para la detección.
  + **RF3.2-Detección con línea personalizada:** El sistema debe detectar el cruce de vehículos sobre una línea definida por el usuario.
* **RF4-Localización de facultad:** La aplicación debe identificar la facultad correspondiente al vídeo mediante la extracción de metadatos.
  + **RF4.1-Extracción automática de metadatos:** El sistema debe extraer las coordenadas GPS del vídeo si están disponibles.
  + **RF4.2-Selección manual de facultad**: Si el vídeo no tiene metadatos, el usuario debe poder seleccionar una facultad manualmente.
* **RF5-Registro en base de datos:** La aplicación debe almacenar en una base de datos todos los datos relativos a las detecciones.
  + **RF5.1-Almacenar fecha y hora:** Debe registrar la fecha y hora en la que se ha realizado la detección.

*Apéndice B. Especificación de requisitos*

* + **RF5.2-Almacenar tipo de vehículo:** Debe registrar el tipo de vehículo detectado.
  + **RF5.3-Almacenar modelo utilizado:** Debe registrar el modelo de YOLOv11 utilizado.
  + **RF5.4-Almacenar facultad asociada:** Debe registrar a qué facultad corresponde el vídeo.
  + **RF5.5-Almacenar dispositivo de procesamiento:** Debe registrar si el vídeo se ha procesado con CPU o GPU.
  + **RF5.6-Almacenar nombre del archivo:** Debe registrar el nombre del archivo cargado por el usuario.
* **RF6-Visualización de resultados:** La aplicación debe mostrar los resultados de las detecciones en forma de gráficos y estadísticas.
  + **RF6.1-Gráfico de barras:** Debe mostrar gráficamente el número de vehículos por día y tipo.
  + **RF6.2-KPIs generales:** Debe presentar métricas, estadísticas y porcentajes de cada tipo de vehículo.
  + **RF6.3-Tabla interactiva:** Debe mostrar una tabla dinámica con los datos del análisis, permitiendo buscar y ordenar los resultados.
* **RF7-Exportación de resultados:** La aplicación debe permitir exportar los resultados de los informes a un archivo CSV.
* **RF8-Internacionalización:** La aplicación debe ofrecer la posibilidad de cambiar el idioma de la interfaz.
* **RF9-Logout de usuarios:** La aplicación debe permitir al usuario cerrar sesión.
* **RF10-Registro de usuarios**: La aplicación debe permitir registrar nuevos usuarios, indicando nombre, contraseña y rol
  + **RF10.1-Almacenar contraseñas:** Debe almacenar las contraseñas de forma segura en la base de datos.
  + **RF10.2-Controlar duplicados:** El sistema debe evitar que se registren nombres de usuarios ya existentes.

*Apéndice B. Especificación de requisitos*

**Requisitos no funcionales**

* **RNF1-Rendimiento:** La aplicación debe realizar la detección en un tiempo razonable, haciendo uso de la GPU cuando esté disponible.
* **RNF2-Disponibilidad:** La aplicación debe ser accesible desde la mayoría de navegadores web.
* **RNF3-Usabilidad:** La aplicación debe ser clara e intuitiva para cualquier usuario, además de adaptarse a diferentes formatos de pantalla.
* **RNF4-Compatibilidad:** La aplicación debe ser compatible con los formatos de vídeo más habituales.
* **RNF5-Escalabilidad:** La aplicación debe permitir desarrollos futuros que generen mayor cantidad de datos y funcionalidades.
* **RNF6-Mantenibilidad:** La aplicación debe estar estructurada adecuadamente para facilitar su mantenimiento.
* **RNF7-Internacionalizaion:** La aplicación debe soportar varios idiomas.
* **RNF8-Integridad de los datos:** La aplicación debe registrar en la base de datos la información adecuada.

*Apéndice B. Especificación de requisitos*

## Especificación de requisitos

**Casos de uso**

|  |  |
| --- | --- |
| **CU-01** | **Login de usuarios** |
| **Requisitos relacionados** | RF1 |
| **Descripción** | Permite al usuario acceder a la aplicación. |
| **Precondiciones** | El usuario sabe las credenciales. |
| **Acciones** | 1. El usuario accede al formulario de login. 2. El usuario introduce usuario y contraseña. 3. Pulsa el botón de iniciar sesión. |
| **Postcondiciones** | Acceso a la aplicación con el rol adecuado. |
| **Excepciones** | Usuario o contraseña incorrectos. (mensaje) |
| **Importancia** | Alta. |

Tabla B.1: CU01 – Login de usuarios

|  |  |
| --- | --- |
| **CU-01** | **Subida y configuración de vídeos** |
| **Requisitos relacionados** | RF2 |
| **Descripción** | Permite al usuario experto subir un vídeo, seleccionar el modelo y dispositivo. |
| **Precondiciones** | El usuario debe estar autenticado con el rol “experto” y no debe haber ningún procesamiento en curso. |
| **Acciones** | 1. Se selecciona un modelo. 2. Se selecciona el tipo de dispositivo. 3. Se selecciona el archivo de vídeo. 4. Pulsa subir vídeo. |
| **Postcondiciones** | Vídeo listo para ser procesado. |
| **Excepciones** | Formato no soportado, intento de subir vídeo sin finalizar procesamiento. |
| **Importancia** | Alta. |

Tabla B.2: CU02 – Subida y configuración de vídeos

*Apéndice B. Especificación de requisitos*

|  |  |
| --- | --- |
| **CU-03** | **Detección y conteo de vehículos** |
| **Requisitos relacionados** | RF3, RF3.1, RF3.2 |
| **Descripción** | Permite detectar y contar vehículos automáticamente. |
| **Precondiciones** | El vídeo ha sido subido. |
| **Acciones** | 1. El sistema analiza el vídeo. 2. El usuario dibuja la línea de conteo. 3. El sistema comienza el procesamiento. 4. El usuario visualiza el procesamiento en tiempo real. 5. El usuario pulsa el botón de “Finalizar procesamiento” para poder cargar otro vídeo o darlo por concluido. |
| **Postcondiciones** | Se obtiene el número de vehículos contado. |
| **Excepciones** | Intento de carga de nuevo vídeo sin finalizar procesamiento, no se detectan vehículos. |
| **Importancia** | Alta. |

Tabla B.3: CU03 – Detección y conteo de vehículos

|  |  |
| --- | --- |
| **CU-04** | **Identificación de facultad** |
| **Requisitos relacionados** | RF4, RF4.1, RF4.2 |
| **Descripción** | Permite asociar una facultad al vídeo seleccionado. |
| **Precondiciones** | El vídeo ha sido subido. |
| **Acciones** | 1. El sistema extrae los metadatos del vídeo. 2. Si hay coordenadas, asocia la facultad correspondiente. 3. Si no hay coordenadas, se solicita al usuario que seleccione una facultad. |
| **Postcondiciones** | Facultad asignada correctamente. |
| **Excepciones** | Metadatos en formato nuevo o incorrecto. |
| **Importancia** | Alta. |

Tabla B.4: CU04 – Identificación de facultad

*Apéndice B. Especificación de requisitos*

|  |  |
| --- | --- |
| **CU-05** | **Registro de logs en base de datos** |
| **Requisitos relacionados** | RF5, RF5.1, RF5.2, RF5.3, RF5.4, RF5.5, RF5.6 |
| **Descripción** | Guarda en la base de datos los logs del análisis del vídeo. |
| **Precondiciones** | El vehículo se ha detectado y contado correctamente. |
| **Acciones** | Se almacenan los campos de la detección en la base de datos. |
| **Postcondiciones** | Datos registrados en la base de datos. |
| **Excepciones** | Fallo de conexión con la base de datos. |
| **Importancia** | Alta. |

Tabla B.5: CU05 – Registro de logs en base de datos

|  |  |
| --- | --- |
| **CU-06** | **Visualización de resultados** |
| **Requisitos relacionados** | RF6, RF6.1, RF6.2, RF6.3 |
| **Descripción** | Permite consultar los resultados de la aplicación de un periodo. |
| **Precondiciones** | El usuario debe haber iniciado sesión como “user” y debe haber datos en la base de datos. |
| **Acciones** | 1. Usuario selecciona periodo de consulta. 2. Se pulsa “Generar informe”. 3. Se visualiza el gráfico de barras correspondiente. 4. Se visualizan los KPIs.. 5. Se visualiza la tabla interactiva. |
| **Postcondiciones** | Los resultados se visualizan correctamente. |
| **Excepciones** | No hay resultados en el periodo seleccionado. |
| **Importancia** | Alta. |

Tabla B.6: CU06 – Visualización de resultados

*Apéndice B. Especificación de requisitos*

|  |  |
| --- | --- |
| **CU-07** | **Exportación de resultados** |
| **Requisitos relacionados** | RF7 |
| **Descripción** | Permite exportar los resultados de los informes a un archivo CSV. |
| **Precondiciones** | El usuario está registrado como “user” y ha seleccionado un periodo válido. |
| **Acciones** | 1. El usuario pulsa el botón “Exportar CSV”. 2. El sistema genera y descarga el archivo CSV. |
| **Postcondiciones** | Archivo CSV generado y descargado correctamente. |
| **Excepciones** | Error en la generación/descarga. |
| **Importancia** | Media. |

Tabla B.7: CU07 – Exportación de resultados

|  |  |
| --- | --- |
| **CU-08** | **Internacionalización** |
| **Requisitos relacionados** | RF8 |
| **Descripción** | Permite cambiar el idioma de la interfaz de la aplicación. |
| **Precondiciones** | El usuario debe haber iniciado sesión. |
| **Acciones** | El usuario selecciona el idioma deseado. |
| **Postcondiciones** | Interfaz mostrada en el idioma elegido. |
| **Excepciones** | Idioma no disponible. |
| **Importancia** | Baja. |

Tabla B.8: CU08 – Internacionalización

*Apéndice B. Especificación de requisitos*

|  |  |
| --- | --- |
| **CU-09** | **Logout de usuarios** |
| **Requisitos relacionados** | RF9 |
| **Descripción** | Permite al usuario cerrar sesión para poder logearse con otro. |
| **Precondiciones** | El usuario ha iniciado sesión. |
| **Acciones** | 1. El usuario pulsa el botón “Cerrar sesión”. 2. El usuario es redirigido a la pantalla de login. |
| **Postcondiciones** | El usuario debe autenticarse de nuevo. |
| **Excepciones** | Si la sesión ha expirado, se muestra la pantalla de login. |
| **Importancia** | Media. |

Tabla B.9: CU09 – Logout de usuarios

|  |  |
| --- | --- |
| **CU-10** | **Registro de usuarios** |
| **Requisitos relacionados** | RF10, RF10.1, RF10.2 |
| **Descripción** | Permite registrar nuevos usuarios en la aplicación. |
| **Precondiciones** | El usuario no está autenticado y desea crear una cuenta. |
| **Acciones** | 1. El usuario pulsa el botón “Cerrar sesión”. 2. El usuario es redirigido a la pantalla de login. |
| **Postcondiciones** | Usuario registrado correctamente en la base de datos. |
| **Excepciones** | Nombre de usuario ya existente. |
| **Importancia** | Alta |

Tabla B.10: CU10 – Registro de usuarios

*Apéndice B. Especificación de requisitos*

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura B.1: Diagrama de casos de uso

*Apéndice C*

# Especificación de diseño

## Introducción

En esta sección se describe la organización y diseño del software desarrollado.

## Diseño de datos

La aplicación cuenta con una base de datos MySQL compuesta por dos tablas principales, diseñadas para almacenar tanto la información de los vehículos detectados como la gestión de usuarios registrados en el sistema. Esta separación permite una mejor organización de los datos y facilita tanto la administración de accesos como la explotación de los resultados.

La tabla **vehicle\_logs** almacena un registro detallado de cada vehículo detectado y cuenta con los siguientes campos:

* timestamp: Fecha y hora de la detección.
* vehicle\_type: Tipo de vehículo detectado.
* model\_used: Versión de YOLOv11 utilizado.
* facultad: Facultad asignada.
* device\_used: Dispositivo utilizado.
* video\_filename: Nombre del archivo procesado.

*Apéndice C. Especificación de diseño*

Tabla

El contenido generado por IA puede ser incorrecto.

Figura C.1: Estructura tabla vehículos

La tabla **users** permite gestionar los usuarios que pueden acceder al sistema, almacenando sus credenciales y rol asociado.

* username: Nombre de usuario único
* password: Contraseña asociada al usuario (resumida criptográficamente en un hash)
* role: Rol asignado al usuario

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura C.2: Estructura tabla usuarios

*Apéndice C. Especificación de diseño*

## Diseño procedimental

En este apartado se describe el funcionamiento de la aplicación mediante varios diagramas de secuencia.

**Login**

El siguiente diagrama de secuencia muestra el flujo correspondiente al proceso de autenticación de los usuarios.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura C.3: Diagrama de secuencia de login

*Apéndice C. Especificación de diseño*

**Subida y procesamiento de vídeo**

En este diagrama se muestra el flujo que sigue el usuario “experto” para realizar el análisis de un vídeo.

Diagrama, Tabla

El contenido generado por IA puede ser incorrecto.

Figura C.4: Diagrama de secuencia de subida y procesamiento

*Apéndice C. Especificación de diseño*

**Visualización de resultados**

Este diagrama muestra el proceso que sigue un usuario “estándar” para visualizar los datos del análisis.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura C.5: Diagrama de secuencia de visualización de resultados

*Apéndice C. Especificación de diseño*

**Registro de usuarios**

Este diagrama muestra el proceso que sigue cualquier usuario para crear una cuenta con el rol adecuado.

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura C.6: Diagrama de secuencia de registro de usuarios

*Apéndice C. Especificación de diseño*

## Diseño arquitectónico

En este apartado se definen los patrones que se han empleado en el diseño de la aplicación.

**Modelo-Vista-Controlador (MVC) y Fachada**

En el desarrollo del proyecto, se ha seguido la arquitectura Modelo-Vista-Controlador combinado con Fachada.

* **Modelo:** Compuesto por los módulos que se encargan de la lógica y la gestión de datos del proyecto. Se incluyen el procesador de detección, el gestor de facultades, el extractor de metadatos y el gestor de la base de datos.
* **Vista:** Formado por las plantillas que conforman la interfaz de usuario. Se ha empleado HTML junto con JavaScript, CSS y Bootstrap para darle formato.
* **Controlador:** Encargado de recibir las solicitudes de la vista y coordinar la lógica del flujo con los módulos del modelo. El modulo app se encarga de las rutas de Flask y coordina la ejecución de la lógica de los modelos.

En el proyecto, el módulo app además de actuar como Controlador en el patrón MVC, también implementa el patrón fachada. Esto significa que todas las operaciones solicitadas desde la vista pasan a través del controlador, que hace de interfaz única. De esta forma, el controlador centraliza la gestión de las peticiones y reenvía a los módulos individuales las tareas específicas sin que la vista y los módulos interactúen entre sí.

Esto facilita el mantenimiento y la validación de datos al concentrar la lógica en un único punto.

*Apéndice C. Especificación de diseño*

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Figura C.7: Patrón MVC [2]

Diagrama

El contenido generado por IA puede ser incorrecto.

Figura C.8: Patrón Fachada [3]

*Apéndice C. Especificación de diseño*

## Diseño de interfaces

Durante las primeras semanas de desarrollo y tras tener una idea clara de cómo se iba a desarrollar el proyecto, se realizó un prototipo de la interfaz de la aplicación con la herramienta Pencil. [4]

Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

Figura C.9: Prototipo de interfaz

*Apéndice C. Especificación de diseño*

Este prototipo resultó muy útil para enfocar el desarrollo de los sprints, permitiendo definir con claridad las funcionalidades de cada rol de usuario

Para el desarrollo de la aplicación final, se empleó una plantilla de Boostrap que facilitó la creación de la interfaz.

Una captura de pantalla de una red social

El contenido generado por IA puede ser incorrecto.

Figura C.10: Interfaz final

*Apéndice D*

# Documentación técnica de programación

## Introducción

En este apartado se describe en detalle cómo está organizado el proyecto, las instrucciones para poder ejecutar la aplicación y los requisitos necesarios.

## Estructura de directorios

El proyecto cuenta con 3 directorios principales:

* **src**: En este directorio se encuentra todo el código necesario para ejecutar el proyecto.
* **vids**: Se encuentran varios vídeos de ejemplo con los que poder probar el funcionamiento de la aplicación.
* **doc**: En este directorio se almacenan los documentos de los anexos y la memoria del proyecto.

**src:**

Dentro de este directorio, podemos encontrar los archivos Python que se encargan del funcionamiento del código. A su vez, podemos encontrar dos subdirectorios:

* **static/icons**: Imágenes empleadas en la aplicación.
* **templates**: Archivos HTML con el diseño de cada página de la interfaz.

*Apéndice D. Documentación técnica de programación*

## Manual del programador

**Instalación de Python**

Para ejecutar el proyecto, es necesario tener instalado Python. Al emplear la librería torch, no todas las versiones de Python son compatibles con esta. Para ello, podemos comprobar el listado aquí: <https://pytorch.org/get-started/locally/>

Se puede descargar Python desde el siguiente enlace: <https://www.python.org/downloads/>

**Instalación de MySQL**

Es necesario descargar MySQL Server. El instalador se encuentra en: <https://dev.mysql.com/downloads/>

Podemos descargar a su vez MySQL Workbench para visualizar la interfaz. Se puede descargar seleccionando la opción al ejecutar el instalador anterior, o por separado desde el enlace proporcionado.

Para que la conexión con Python no falle, se recomienda lanzar las siguientes consultas desde Workbench, o en su defecto, realizarlo mediante los botones de la interfaz.

*CREATE DATABASE IF NOT EXISTS vehicle\_detection;*

*CREATE USER IF NOT EXISTS 'Usuario'@'localhost' IDENTIFIED BY 'Contraseña'*

*GRANT ALL PRIVILEGES ON vehicle\_detection.\* TO 'Usuario'@'localhost';*

*FLUSH PRIVILEGES;*

Esto creará la base de datos, y se le asignará un usuario que tenga acceso. Es **importante** recordar el usuario y contraseña introducidos en estos pasos, ya que se necesitarán más adelante.

*Apéndice D. Documentación técnica de programación*

**Instalación de PyTorch CUDA**

El modelo de detección YOLO emplea la librería torch. Para mejorar el rendimiento de este, se puede instalar la librería torch con soporte de CUDA, solamente válido para dispositivos con GPU de Nvidia. Esto hará que el modelo mejoré en cuanto a rendimiento gracias a la aceleración por GPU. Es **obligatorio** tener actualizados los drivers de nuestra tarjeta gráfica.

Para instalar esta versión, se puede hacer desde terminal o modificando el requirements.txt con la versión adecuada. El comando se puede encontrar en el siguiente enlace, y se deberá seleccionar la plataforma CUDA adecuada: <https://pytorch.org/get-started/locally/>

Para comprobar la última versión de CUDA compatible con nuestra gráfica, se puede realizar desde la terminal mediante el comando:

*nvidia-smi*

Para asegurarnos de que la librería cuenta con CUDA, se puede comprobar con:

*python*

*import torch*

*print("CUDA disponible:", torch.cuda.is\_available())*

Si esto último devuelve true, torch está instalado correctamente. En ocasiones, puede que sea necesario instalar el soporte cuDDN para que funcione correctamente. Se puede descargar desde: <https://developer.nvidia.com/rdp/cudnn-archive>

En caso de no contar con GPU de Nvidia, no hace falta instalar *torch* ya que la librería *ultralytics* la instala por defecto. Sería necesario comentar las líneas de las librerías *torch, torchaudio* y *torchvision.*

*Apéndice D. Documentación técnica de programación*

## Compilación, instalación y ejecución del proyecto

Para la ejecución del proyecto, debemos clonar el repositorio de GitHub que se encuentra en <https://github.com/pabloosp/VehicleDetection>

Para ello, lanzamos el comando que aparece desde el directorio deseado o descargamos el archivo zip, lo descomprimimos y lo movemos de directorio si queremos. A continuación, abrimos la carpeta con Visual Studio Code o el IDE que empleemos. Seleccionamos *File>Open Folder* y la carpeta del proyecto.

**Librerías de Python**

Es necesario instalar una serie de librerías para que el proyecto funcione correctamente. Se recomienda instalar un entorno virtual e instalar las librerías desde este para que no haya problemas con otros proyectos que tengamos. Se puede encontrar más información en: <https://docs.python.org/es/3.13/tutorial/venv.html>

Se proporciona el archivo *requirements.txt* que contiene todas las librerías necesarias. Si hemos instalado previamente una versión de *torch* diferente a la de este archivo, será necesario adaptar las versiones de *torch*, *torchaudio* y *torchvision* o en su defecto, comentarlas. Lanzamos el siguiente comando para instalar todas las librerías, este paso puede tardar unos minutos.

*pip install -r requirements.txt*

Si se prefiere, se pueden instalar las librerías por separado, mediante el comando *pip install* seguido de la librería. Las librerías necesarias son: *flask, python-dotenv, mysql-connector-python, opencv-python, ultralytics, shapely, pymediainfo, lapx, Jinja2*. También es necesario *torch*, pero la librería variará dependiendo de si se quiere instalar la versión CUDA o no, tal y como se ha indicado anteriormente.

**Variables de entorno**

Para que la aplicación pueda leer de forma segura las credenciales de la base de datos y la clave de Flask, se emplea un fichero .env. En la raíz del proyecto descargado, podemos ver un archivo llamado .env.example. Lo copiamos y lo renombramos a .env. También se puede realizar mediante comandos:

*cp .env.example .env*

*Apéndice D. Documentación técnica de programación*

A continuación, debemos editar el archivo .env, y rellenar con los valores adecuados.

* FLASK\_SECRET 🡪 Clave para la sesión de Flask. Puede ser cualquiera, pero se recomienda lanzar el siguiente comando para obtener una clave segura.

*python -c "import secrets; print(secrets.token\_hex(16))"*

* MYSQL\_USER 🡪 Usuario creado en el apartado *D.3.*
* MYSQL\_PASSWORD 🡪 Contraseña creada en el apartado *D.3.*

**Ejecución del proyecto**

Una vez realizado correctamente lo anterior, ya se puede ejecutar la aplicación. Para ello, una vez estemos en el directorio del proyecto, lanzamos los siguientes comandos:

*cd src*

*py app.py*

La primera vez que lancemos la aplicación, tardará unos minutos ya que se descargará el modelo YOLOv11 Small.

Una vez se haya lanzado correctamente, se podrá ver por terminal que todas las conexiones son correctas. Mientras se utilice la aplicación, no se debe cerrar la terminal. Se podrá acceder al proyecto en <http://localhost:5000/>

También, la primera vez que se ejecuten las diferentes versiones de YOLOv11, se descargarán automáticamente por la terminal.

Texto

El contenido generado por IA puede ser incorrecto.

Figura D.1: Conexiones correctas

**Máquina Virtual**

Si se utiliza la máquina virtual proporcionada, la clave de acceso es *tfg.*

*Apéndice D. Documentación técnica de programación*

## Pruebas del sistema

Para garantizar el correcto funcionamiento de la aplicación, se han realizado diferentes pruebas. Estas verifican que los diferentes componentes del proyecto funcionan según lo esperado.

Desde el inicio de la ejecución del proyecto, el sistema realiza comprobaciones para verificar que la conexión con la base de datos es correcta, las tablas se han creado correctamente y el archivo .env se ha cargado sin fallos. Para ello, se han empleado bloques try-except que permiten detectar errores y lanzar una excepción en caso de que alguna comprobación falle. Estas se muestran en la terminal de forma visual, y nos permiten aplicar las medidas necesarias para corregir los posibles errores sin detener el sistema. También se han empleado en diferentes módulos como la carga de vídeos, el procesamiento, la exportación de resultados.

Texto

El contenido generado por IA puede ser incorrecto.

Figura D.2: Ejemplo bloque try-except

Además, se ha realizado un análisis de la calidad del código. Se ha empleado la herramienta SonarCloud que permite realizar un análisis completo del código fuente. Los principales resultados del análisis han sido:

* Security: 0 issues. Clasificación A.
* Reliability: 2 issues. Clasificación A.
* Mantainability: 7 issues. Clasificación A.
* Duplications: 6,4%
* Security hotspots: 0

*Apéndice D. Documentación técnica de programación*

Para conseguir estos resultados, fue necesario arreglar pequeños errores en cuanto al manejo de datos para conseguir arreglar todos los problemas del apartado de seguridad. En cuanto a reliability, se pueden observar 2 issues sin mayor importancia. En maintainability, se han detectado 7 issues relacionados principalmente con código comentado y duplicado pero intencional. El porcentaje de código duplicado puede considerarse bajo, y la mayor parte tiene que ver con el botón de internacionalización.

El análisis detallado se puede encontrar en:

<https://sonarcloud.io/project/overview?id=pabloosp_VehicleDetection>

La conclusión obtenida es que, gracias a los diferentes mecanismos de prueba implementados, se ha comprobado que el proyecto es robusto, mantenible y seguro. Las medidas empleadas permiten anticipar posibles fallos y minimizar su impacto.

*Apéndice E*

# Documentación de usuario

## Introducción

En este apéndice se explica cómo usar la aplicación por parte del usuario.

## Requisitos de usuarios

Para utilizar la aplicación, el usuario solamente debe contar con un dispositivo que tenga una versión reciente de un navegador web:

* Google Chrome
* Mozilla Firefox
* Opera
* Safari
* Brave

## Instalación

El proyecto no se ha podido alojar en ninguna web de hosting gratuita. Por ello, el usuario deberá seguir las instrucciones de instalación del manual del programador, o en su defecto, ejecutar la máquina virtual proporcionada. De ambas maneras, se accederá con <http://localhost:5000/>

*Apéndice E. Documentación de usuario*

## Manual del usuario

**Login**

La primera ventana que se muestra al entrar en la aplicación es la de iniciar sesión. Si es la primera vez que se utiliza, debemos pulsar el botón **Registrarse** y nos redirigirá a la ventana de registro.

En esta ventana, una vez creada la cuenta, introduciremos el usuario y contraseña que hayamos introducido al registrarnos. Al pulsar el botón **Entrar**, El sistema nos redirige automáticamente dependiendo del rol que hayamos elegido.

En caso de que la cuenta no exista o los datos introducidos sean incorrectos, se mostrará un aviso indicándolo.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura E.1: Página de inicio

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Figura E.2: Acceso denegado

*Apéndice E. Documentación de usuario*

**Registro**

Tras haber seleccionado el botón **Registrarse** en la ventana de Login, se muestra una ventana donde podemos crear una cuenta. Es necesario rellenar todos los campos:

* Usuario: Nombre de la cuenta, no debe existir previamente.
* Contraseña: Sin requisitos, se genera un resumen criptográfico (hash) al guardarse en la base de datos.
* Rol: Rol del usuario de la cuenta. Usuario se encarga de la visualización y explotación de datos mientras que Experto se encarga de la subida y procesamiento de vídeos.

Una vez introducidos estos campos, pulsamos el botón **Registrarse** y si la cuenta se ha creado correctamente, nos mostrará un aviso.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura E.3: Página de registro

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Figura E.4: Usuario registrado correctamente

En caso de que hayamos entrado a está página por error, podemos pulsar el botón **Volver al login** y se nos redirigirá nuevamente a la ventana de Login.

*Apéndice E. Documentación de usuario*

**Menú de procesamiento**

Cuando se inicia sesión con las credenciales de un usuario “experto”, se nos redirigirá a la ventana de este. Esta ventana cuenta con diferentes opciones que el usuario puede seleccionar a su gusto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura E.5: Ventana usuario con rol experto

En la columna de la izquierda, se puede observar un recuadro con una serie de opciones. Seleccionar modelo nos permite elegir el modelo de YOLOv11 con el que se va a procesar el vídeo. Están ordenados de menor a mayor, siendo Nano el más rápido, pero menos preciso y Large el más lento pero de mayor precisión.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura E.6: Selección de modelo

*Apéndice E. Documentación de usuario*

A continuación, nos dice si queremos usar GPU o CPU. Esta opción solo está disponible si el dispositivo desde el que estamos ejecutando la aplicación cuenta con una tarjeta gráfica de Nvidia. Seleccionamos la opción deseada.



Figura E.7: GPU activada



Figura E.8: GPU desactivada

Una vez seleccionado lo deseado, pulsaremos el botón **Seleccionar archivo** se abrirá el explorador de archivos de nuestro dispositivo. Deberemos seleccionar el archivo a procesar. Se admite cualquier formato de vídeo.

Pulsamos el botón **Subir Vídeo**, si se nos ha olvidado seleccionar alguna de las opciones anteriores saltará el aviso correspondiente. Una vez pulsado el botón y dependiendo del vídeo subido, hay diferentes formas de proceder.

**Vídeos con metadatos y facultad detectada**

El sistema ha podido leer los metadatos del vídeo. Se han extraído las coordenadas y ha detectado que el vídeo ha sido grabado en una de las facultades de la Universidad de Burgos. Dependiendo de las coordenadas, el sistema sabe a qué facultad pertenece el vídeo.

*Apéndice E. Documentación de usuario*

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Figura E.9: Ventana caso 1

*Apéndice E. Documentación de usuario*

Se puede observar que el sistema nos dice que la facultad ha sido detectada automáticamente y que el vídeo en este caso pertenece a EPS Vena. A continuación, nos muestra un resumen de las opciones que el usuario ha elegido para procesar el vídeo. Se observa a su vez el primer frame del vídeo subido. En este recuadro deberemos dibujar la línea de cruce para que el sistema realice el conteo de vehículos. Pulsamos el click izquierdo del ratón, desplazamos hasta donde queramos dibujar la línea y soltamos. La línea se habrá dibujado y solo quedará pulsar **Confirmar línea** para comenzar con el procesamiento.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

Figura E.10: Línea dibujada

*Apéndice E. Documentación de usuario*

Antes de esto, podremos ver las coordenadas que el sistema ha detectado para asegurarnos que la asignación de facultad es correcta.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura E.11: Coordenadas detectadas

Una vez pulsemos el botón **Confirmar línea**, el procesamiento comenzará. El recuadro de la derecha que previamente estaba en negro mostrará el procesamiento del vídeo subido en tiempo real. Se podrá ver un rectángulo verde alrededor de los coches que están siendo detectados, además de un punto rojo en el centro del recuadro. También se observará la línea dibujada previamente y un contador indicando los vehículos contados hasta el momento. Todos los datos de los vehículos contados se almacenan en la base de datos.

Forma

El contenido generado por IA puede ser incorrecto.

Figura E.12: Vídeo sin procesarse

*Apéndice E. Documentación de usuario*

Imagen que contiene pequeño, pasto, juguete, equipaje

El contenido generado por IA puede ser incorrecto.

Figura E.13: Vídeo procesándose

*Apéndice E. Documentación de usuario*

Es posible detener el procesamiento antes de que el vídeo acabe. Para ello, deberemos pulsar el botón **Finalizar Procesamiento.** Los vehículos registrados hasta este momento se habrán guardado correctamente en la base de datos.



Figura E.14: Botón finalizar procesamiento

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Figura E.15: Procesamiento finalizado

**Vídeos sin metadatos o los metadatos no pertenecen a ninguna facultad**

La manera de proceder en ambos casos es la misma, lo único diferente es el mensaje de aviso. El sistema nos dice que el vídeo no contiene metadatos o sí contiene, pero están fuera del rango de cualquiera de las facultades de la Universidad de Burgos.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Figura E.16: Vídeo con metadatos fuera de rango

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Figura E.17: Vídeo sin metadatos

*Apéndice E. Documentación de usuario*

En este caso, se nos mostrará un formulario y deberemos seleccionar manualmente la facultad donde se grabó el vídeo, ya que el sistema no cuenta con la suficiente información para detectarlo por sí solo.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura E.18: Formulario de facultades

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Figura E.19: Listado de facultades

En el caso del vídeo con metadatos fuera de rango, se podrán ver las coordenadas donde fue grabado para asegurarse de que es correcto que están fuera de rango.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Figura E.20: Coordenadas vídeo fuera de rango

*Apéndice E. Documentación de usuario*

Una vez seleccionada la facultad, deberemos pulsar el botón **Asignar Facultad.** A partir de aquí, debemos proceder de igual manera que en el caso de **Vídeos con metadatos y facultad detectada.** Se mostrará el frame para dibujar la línea y se deberá pulsar **Confirmar línea** para que el procesamiento comience.

**Menú de visualización**

Cuando se introducen las credenciales de un usuario con rol “usuario”, se le redirigirá a la ventana adecuada. En esta ventana, podemos ver un selector de fecha de inicio y fin y el botón **Generar Informe.** El informe nos mostrará los datos del periodo seleccionado.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Figura E.21: Selector de fechas

*Apéndice E. Documentación de usuario*

En este punto, seleccionamos el periodo deseado. Una vez hecho, pulsaremos el botón **Generar Informe** y se nos mostrará diferente información.

Lo primero que se puede observar es un gráfico de vehículos por día. En el eje X podemos ver todos los días del periodo. El límite es de 7 días, por lo que si queremos ver los datos de otro día deberemos pulsar el botón **Día anterior** o **Día siguiente** respectivamente. En el eje Y podemos ver el total de coches, y el color dentro de cada barra indica el tipo de vehículo registrado.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Figura E.22: Gráfico de barras

Lo siguiente que se puede observar es un recuadro que nos muestra estadísticas del periodo seleccionado. Este recuadro está dividido a su vez en tres columnas:

*Apéndice E. Documentación de usuario*

* **Resumen general:** Muestra el total de vehículos detectados en el periodo, la media de vehículos por día y los días con más y menos tráfico.

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Figura E.23: Resumen general

* **Porcentajes por tipo:** Indica el porcentaje y número de cada tipo de vehículo.

Interfaz de usuario gráfica, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Figura E.24 : Porcentajes por tipo

*Apéndice E. Documentación de usuario*

* **Estadísticas técnicas:** Muestra el modelo de YOLOv11 más usado, cuantos vídeos distintos se han procesado y la facultad más y menos activa.

Texto

El contenido generado por IA puede ser incorrecto.

Figura E.25 : Estadísticas técnicas

Lo último que se puede observar es una tabla donde se muestran todas las facultades con registros y el total de vehículos detectados con su correspondiente desglose. Los valores de esta tabla se pueden ordenar y se pueden buscar datos exactos.

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

Figura E.26 : Tabla de facultades

*Apéndice E. Documentación de usuario*

Al final de esta ventana, hay un botón **Exportar CSV** que descargará todos los datos del periodo seleccionado para que el usuario pueda realizar la explotación de la información que considere.



Figura E.27 : Botón Exportar CSV

**Cierre de sesión**

Tanto el **Menú de procesamiento** como el **Menú de visualización** cuentan con un botón **Cerrar sesión.** Al pulsar el botón, se nos redirigirá a la ventana de Login.



Figura E.28 : Botón Cerrar sesión

**Cambiar idioma**

Todas las ventanas cuentan con un botón que nos permite cambiar el idioma de la aplicación. La aplicación está disponible tanto en inglés como en español.

Imagen de la pantalla de un celular con letras

El contenido generado por IA puede ser incorrecto.

Figura E.29 : Botón cambiar idioma

*Apéndice F*

# Anexo de sostenibilización curricular

## Introducción

Este proyecto surge como respuesta a la necesidad de reducir el impacto ambiental derivado del tráfico de vehículos en los campus universitarios. A través del desarrollo del proyecto, se busca mejorar la eficiencia en los desplazamientos, optimizar la gestión de espacios y fomentar un uso más responsable del transporte. Esto se alinea con los objetivos de sostenibilidad promovidos por el ranking Greenmetrics [5], que valora el uso de herramientas que midan el impacto ambiental en las universidades.

## Competencias en sostenibilidad adquiridas

**Aplicación de tecnologías sostenibles:**

Se ha implementado una solución que prescinde de sensores físicos, reduciendo así el consumo de materiales y energía. La visión por computador permite aprovechar recursos por lo que se refuerza el enfoque sostenible.

**Responsabilidad ambiental:**

El proyecto ha sido ideado desde un punto de vista de compromiso con la mejora de nuestro alrededor. La recopilación y análisis de datos no persigue únicamente un fin técnico, sino que busca plantear políticas y herramientas que ayuden a reducir las emisiones y mejoren la calidad de vida en los campus.

*Apéndice F. Anexo de sostenibilización curricular*

**Análisis crítico:**

El desarrollo del proyecto ha abierto puertas en cuanto a reflexionar sobre el uso excesivo del vehículo privado. A su vez, facilita la evaluación de posibles medidas como el fomento del transporte público o el uso de la bicicleta.

## Impacto social y ambiental

**Impacto social:**

El análisis de patrones de movilidad permite adaptar las decisiones a las necesidades reales de la comunidad universitaria. Mejorar la gestión y organización del tráfico, promover métodos de transporte alternativos o establecer ciertas restricciones pueden beneficiar tanto a estudiantes como a los docentes y personal administrativo.

**Impacto ambiental:**

La reducción del tráfico implica una menor emisión de gases contaminantes, menor ruido y una disminución del consumo energético. Al obtener una visión detallada del uso real de acceso a los edificios de las diferentes facultades, se facilita la implementación de medidas y estrategias sostenibles en cuanto a la gestión del espacio y la movilidad.

## Conclusión

Este proyecto muestra como la tecnología puede ser una gran herramienta para el cambio orientado hacia objetivos medioambientales. La aplicación de la visión por computador en el estudio de la movilidad representa un paso hacia una gestión más eficiente y sostenible. Además, contribuye a posicionar a la Universidad de Burgos en el ranking Greenmetrics, mostrando un modelo de universidad comprometida con el desarrollo sostenible y el respeto por el medio ambiente.

# Bibliografía

[1] «Seguridad Social: Cotización / Recaudación de Trabajadores». Accedido: 19 de mayo de 2025. [En línea]. Disponible en: https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/36537

[2] «Modelo–vista–controlador», *Wikipedia, la enciclopedia libre*. 7 de marzo de 2025. Accedido: 22 de mayo de 2025. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Modelo%E2%80%93vista%E2%80%93controlador&oldid=165920571

[3] «GoF Design Pattern Template: Facade», Visual Paradigm Community Circle. Accedido: 22 de mayo de 2025. [En línea]. Disponible en: https://circle.visual-paradigm.com/facade/

[4] «Home - Pencil Project». Accedido: 22 de mayo de 2025. [En línea]. Disponible en: https://pencil.evolus.vn/

[5] «Overall Rankings 2024 - UI GreenMetric». Accedido: 4 de junio de 2025. [En línea]. Disponible en: https://greenmetric.ui.ac.id/rankings/overall-rankings-2024