



**Universidad
Europea**

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERIA INFORMÁTICA

PROYECTO COMPUTACIÓN I

TrafiVision

Grupo 5

Anyi Lin

Rodrigo DÍAZ MARTÍNEZ

Dirigido por

MIQUEL COLOBRAN HUGUET

CURSO 2025-2026

TÍTULO: TrafiVisión

AUTOR: Anyi Lin,Rodrigo DÍAZ MARTÍNEZ

TITULACIÓN: GRADO EN INGENIERÍA INFORMÁTICA

DIRECTOR/ES DEL PROYECTO: MIQUEL COLOBRAN HUGUET

FECHA: Enero 2026

REPOSITORIO: <https://github.com/RodrigoVII/TrafiVison>

RESUMEN

El resumen tiene entre 150-250 palabras. Resumir consiste en ofrecer información sobre cómo, dónde, cuándo y por qué se aplica el proyecto. Se realiza al finalizar el trabajo.

Debe incluir:

- Resumen del problema planteado y las aportaciones más importantes del proyecto al respecto
- Indica si procede si el proyecto se realizó en colaboración con una empresa, indicando nombre (siempre que haya autorización expresa por la empresa), y el sector industrial o empresarial
- Principales resultados del proyecto (diseño de un sistema para..., desarrollo e implementación de una solución para..., análisis de..., modelo de...)
- Resume las conclusiones más importantes del trabajo realizado

Palabras clave: hasta un máximo de 6 conceptos (un concepto puede conllevar una o más palabras). Incluye tecnologías que hayas utilizado, conceptos relevantes del ámbito científico-técnico, y conceptos de la industria (algunos ejemplos: machine learning, Big Data, Smart City, visión artificial, social media, Twitter, computación afectiva, transformación digital, GDPR, DevOps, EEG, ...)

ABSTRACT

Resumen en inglés.

Keywords: Palabras clave en inglés

INDICE

RESUMEN.....	3
ABSTRACT.....	4
Capítulo 1. INTRODUCCIÓN.....	9
1.1 Estado del arte.....	10
1.2 Contexto y justificación.....	11
1.3 Objetivos generales.....	11
1.4 Objetivos específicos.....	13
1.5 Planificación.....	14
Capítulo 2. TECNOLOGÍAS USADAS.....	15
Capítulo 3. ANÁLISIS DEL PROBLEMA.....	18
Análisis del problema.....	20
Capítulo 4. PROPUESTA DE SOLUCIÓN (DISEÑO).....	25
4.1 Organización de las subtareas.....	25
Organización de subtareas.....	26
Capítulo 5. DESARROLLO / IMPLEMENTACIÓN.....	31
Capítulo 6. Modelado.....	32
Modelado.....	32
Capítulo 7. BIBLIOGRAFÍA/WEBGRAFÍA.....	36
Capítulo 8. ANEXOS.....	38
ANEXO I – Criterios de clasificación y variables derivadas.....	38

Índice de Figuras

Todas las figuras incluidas en la memoria deben estar referenciadas para incluir en este índice.

Índice de Tablas

Todas las tablas incluidas en la memoria deben estar referenciadas para incluir en este índice.

Capítulo 1. INTRODUCCIÓN

El proyecto **TrafiVison** tiene como objetivo construir un sistema automatizado de recopilación y análisis de datos de tráfico en la ciudad de Madrid. El enfoque principal se basa en la creación de un **proceso ETL (Extracción, Transformación y Carga)** capaz de integrar información visual procedente de cámaras públicas del Ayuntamiento de Madrid con datos meteorológicos obtenidos a través de **Open-Meteo**.

En la fase actual, el sistema ya es capaz de **extraer imágenes en tiempo real** de distintas ubicaciones de la ciudad, **registrar su fecha y hora, y obtener el clima correspondiente** (temperatura, precipitación y tipo de lluvia). A partir de estas imágenes, se aplica el modelo de visión por computador **YOLOv8**, que permite **contar vehículos** y clasificar el nivel de tráfico como *bajo, medio o alto*. Todos estos datos se consolidan automáticamente en un conjunto de datos unificado y limpio (CSV), que servirá de base para el análisis posterior.

El trabajo se centra actualmente en garantizar la **calidad, estructura y coherencia temporal** del dataset, corrigiendo incidencias encontradas durante las primeras pruebas (por ejemplo, errores en el formato de hora o pequeños desfases entre la captura de clima e imágenes). Una vez finalizada esta etapa de recopilación y limpieza, se prevé avanzar hacia una fase de **análisis exploratorio** y el desarrollo de un **modelo predictivo** capaz de estimar el nivel de tráfico según las condiciones meteorológicas, la franja horaria y el día de la semana.

TrafiVison pretende servir como una base sólida para el estudio del tráfico urbano mediante datos abiertos y herramientas de aprendizaje automático. El proyecto busca demostrar cómo la **automatización y la integración de fuentes heterogéneas** pueden facilitar el análisis y la toma de decisiones en movilidad, y constituye un primer paso hacia un sistema más amplio de **predicción inteligente de tráfico** en entornos urbanos.

1.1 Estado del arte

En los últimos años, diferentes organismos públicos han puesto a disposición de la ciudadanía **datos abiertos sobre movilidad y tráfico urbano**, permitiendo el desarrollo de proyectos que combinan información visual, meteorológica y de sensores para comprender mejor la congestión vial. En el caso de Madrid, el **Ayuntamiento** ofrece un sistema de **cámaras de tráfico en tiempo real**, accesibles de forma pública, y diversas **APIs de datos abiertos** que incluyen información sobre el estado de las vías, incidencias o intensidad de circulación.

Pese a ello, la mayor parte de estos datos se encuentran **dispersos, en formatos heterogéneos y con frecuencias de actualización variables**, lo que dificulta su aprovechamiento conjunto. De manera similar, las fuentes meteorológicas —como las proporcionadas por **Open-Meteo** o **AEMET**— suelen mantenerse separadas del resto de los sistemas de información de tráfico, reduciendo las posibilidades de correlacionar factores ambientales con los niveles de congestión.

Hasta el momento, la mayoría de proyectos existentes se centran en **visualizar tráfico en tiempo real o mostrar cámaras en mapas interactivos**, pero pocos abordan la **automatización del proceso ETL completo** (Extracción, Transformación y Carga) que permita mantener un **dataset limpio y normalizado** de forma continua. Este tipo de integración es esencial para garantizar la **consistencia temporal y la calidad de los datos** cuando se pretende entrenar modelos de aprendizaje automático.

En este contexto, **TrafiVision** surge como una propuesta práctica y académica que busca **automatizar la recogida y estandarización** de datos de tráfico en Madrid, **fusionando imágenes, información meteorológica y variables temporales** (hora, día de la semana, laborable/festivo). En esta primera fase, el sistema se centra en construir un dataset propio y verificable; en etapas futuras, se prevé extenderlo hacia la **predicción de niveles de tráfico y visualización de patrones de movilidad**.

1.2 Contexto y justificación

El análisis de datos abiertos relacionados con el tráfico se ha convertido en una herramienta esencial para **mejorar la movilidad urbana, reducir la congestión y optimizar la planificación del transporte público y privado**. Sin embargo, la información disponible suele proceder de **fuentes heterogéneas**, con distintos formatos, niveles de actualización y estructuras, lo que **complica su integración y análisis conjunto**.

En este contexto, el proyecto **TrafiVison** nace con el propósito de **unificar y automatizar la recopilación de datos de tráfico y clima en la ciudad de Madrid**, creando un flujo ETL (Extracción, Transformación y Carga) que garantice la coherencia y la trazabilidad de la información. El sistema combina imágenes captadas desde cámaras públicas del Ayuntamiento de Madrid con datos meteorológicos en tiempo real de la API **Open-Meteo**, generando un dataset propio, estructurado y limpio, listo para ser utilizado en futuros análisis.

Desde el punto de vista **académico**, TrafiVison representa un **caso práctico de aplicación de técnicas de ingeniería de datos** dentro del ámbito de la inteligencia artificial y el aprendizaje automático, abordando todas las etapas del ciclo de vida de los datos: obtención, limpieza, integración y validación.

En el plano **social y tecnológico**, el proyecto contribuye al **aprovechamiento de los datos abiertos** como herramienta de conocimiento, fomentando la transparencia y la innovación en la gestión de la movilidad urbana. A largo plazo, TrafiVison busca sentar las bases para el desarrollo de **modelos predictivos** que ayuden a anticipar niveles de tráfico y a diseñar estrategias de movilidad más sostenibles y seguras.

1.3 Objetivos generales

El objetivo principal del proyecto **TrafiVison** es **diseñar e implementar un proceso ETL (Extracción, Transformación y Carga)** que integre datos abiertos de tráfico y clima para generar un **dataset propio, limpio y estructurado**, capaz

de servir como base para análisis y modelos predictivos sobre el comportamiento del tráfico urbano en Madrid.

Actualmente, el proyecto ya cuenta con una **fase de extracción y transformación parcialmente implementada**. Se han automatizado los procesos de:

- descarga de imágenes desde **10 cámaras públicas** del Ayuntamiento de Madrid,
- recogida de datos meteorológicos en tiempo real mediante la API **Open-Meteo**, y
- análisis automatizado de imágenes con **YOLOv8**, que permite detectar vehículos y clasificar el nivel de tráfico.

En la fase de **transformación**, se ha desarrollado un proceso de **limpieza, unión y estandarización de datos**, resolviendo incidencias como los errores de formato en la hora o los desajustes temporales entre cámaras y clima.

El **resultado actual** es un dataset funcional y coherente, en formato CSV, que integra imágenes, clima y etiquetas de tráfico.

De cara al futuro, el proyecto se propone:

- **Ampliar el volumen y la diversidad de datos** (más días, diferentes condiciones meteorológicas, horarios y zonas).
- **Incorporar un modelo predictivo** basado en aprendizaje automático para estimar el nivel de tráfico a partir de variables como hora, temperatura, lluvia o tipo de día (laborable/festivo).
- **Diseñar una interfaz de visualización** que permita explorar los resultados y analizar tendencias de congestión.

En conjunto, TrafiVison busca demostrar cómo la combinación de **datos abiertos, visión por computador y técnicas ETL** puede facilitar la creación de

herramientas útiles para el estudio del tráfico urbano y la toma de decisiones informadas sobre movilidad.

1.4 Objetivos específicos

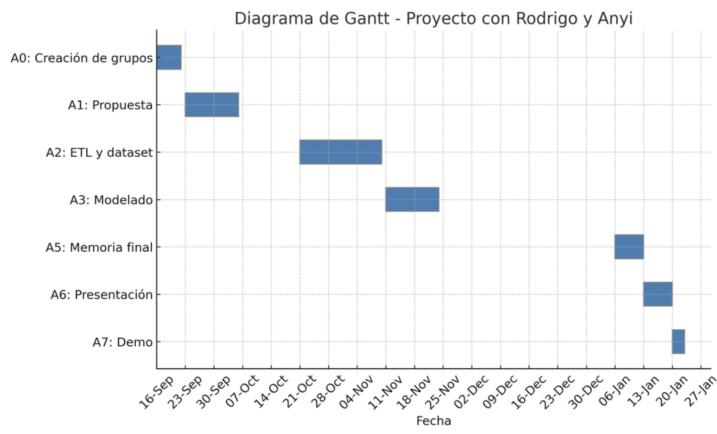
1. **Analizar y evaluar las fuentes de datos abiertas** relacionadas con el tráfico y el clima, priorizando aquellas que sean accesibles, actualizadas y fiables.
En la fase actual se han seleccionado las **cámaras públicas del Ayuntamiento de Madrid** y la **API de Open-Meteo** por su disponibilidad y formato sencillo.
En fases futuras se contempla incorporar otras fuentes complementarias (p. ej., aforos o datos de congestión de portales abiertos).
2. **Desarrollar scripts de extracción automatizados** que permitan obtener los datos de manera periódica sin intervención manual.
Actualmente, el sistema ya captura imágenes y clima cada 15 minutos, sincronizados entre sí.
Se planea extender esta automatización para recopilar datos durante períodos prolongados y habilitar alertas o logs que monitoricen posibles errores de red.
3. **Estandarizar, depurar y enriquecer la información** para asegurar su consistencia y calidad.
Hasta ahora, se han corregido errores de formato (como la “hora fantasma”), eliminado duplicados y añadido variables derivadas como **laborable/no laborable** y **franja horaria**.
En futuras iteraciones se aplicarán métodos de validación adicionales y un control de calidad más exhaustivo para detectar valores anómalos o ausentes.
4. **Unificar la información en un dataset estructurado y documentado**, combinando las distintas fuentes en un único archivo preparado para análisis.

El dataset actual ([dataset_final_limpio.csv](#)) ya permite explorar relaciones entre tráfico, clima y tiempo.

En las próximas fases se elaborará un **diccionario de datos completo**, se incluirán **metadatos descriptivos** y se evaluará la posibilidad de exportar el conjunto en otros formatos (Parquet, SQL o JSON) para facilitar su reutilización.

5. Llevar a cabo análisis exploratorios y crear visualizaciones que permitan identificar patrones en la movilidad y el tráfico.
6. Implementar técnicas básicas de aprendizaje automático para clasificar niveles de congestión o determinar horarios óptimos para el tránsito.

1.5 Planificación



Capítulo 2. TECNOLOGÍAS USADAS

El desarrollo del proyecto **TrafiVison** se apoya en un conjunto de herramientas y tecnologías que permiten automatizar la recopilación, el tratamiento y el análisis de datos de tráfico. A continuación, se describen las principales tecnologías utilizadas hasta el momento y las que se prevé incorporar en fases posteriores.

Tecnologías empleadas actualmente

Python 3.9

Python es el lenguaje principal del proyecto debido a su **versatilidad, amplia comunidad científica** y disponibilidad de librerías orientadas a la manipulación de datos, el aprendizaje automático y la visión por computador.

Todo el flujo ETL del proyecto (extracción, limpieza, unión y análisis de datos) se ha desarrollado íntegramente en Python.

Visual Studio Code

El entorno de desarrollo utilizado es **Visual Studio Code**, por su facilidad de integración con GitHub y su soporte para entornos de Python.

Además, permite ejecutar scripts directamente desde la terminal, organizar carpetas del proyecto y mantener la trazabilidad del código en cada fase.

Bibliotecas de Python

- **requests:** para realizar peticiones HTTP a las fuentes de datos (cámaras del Ayuntamiento y API de Open-Meteo).
- **pandas:** para manipular, limpiar y unir los conjuntos de datos generados en formato CSV.
- **schedule, time y datetime:** para programar tareas automáticas (captura cada 15 minutos) y gestionar fechas y horas.
- **ultralytics (YOLOv8):** librería de visión por computador usada para detectar vehículos en las imágenes. Se seleccionó el modelo **YOLOv8n**

(**nano**) por su equilibrio entre precisión y velocidad.

- **opencv-python**: utilizada de forma indirecta por YOLO para procesar imágenes.
- **torch y torchvision**: bases del modelo YOLO, utilizadas para la inferencia de detección de objetos.

Open-Meteo API

Fuente principal para los datos meteorológicos del proyecto.

Permite obtener temperatura, precipitación y códigos de clima en tiempo real para coordenadas específicas (en este caso, Madrid capital).

Es una API **gratuita y sin autenticación**, lo que facilita la automatización de consultas.

Cámaras del Ayuntamiento de Madrid

Proporcionan imágenes públicas de tráfico actualizadas periódicamente.

El proyecto accede directamente a los enlaces [.jpg](#) de diez cámaras seleccionadas estratégicamente, ubicadas en zonas como Cibeles, Plaza Castilla, Gran Vía o Cuatro Caminos.

Estas imágenes constituyen la base visual del dataset.

Git y GitHub

Se utilizan para **control de versiones** y trabajo colaborativo entre los integrantes del equipo (Rodrigo y Anyi).

El repositorio público [TrafiVison](#) permite almacenar el código, realizar seguimiento de cambios y mantener copias seguras del proyecto.

Conclusión de las tecnologías usadas

El uso de Python y sus bibliotecas especializadas ha permitido **construir un pipeline sólido, modular y reproducible**.

En las fases siguientes, la incorporación de entornos de inteligencia artificial y plataformas de visualización ampliará el alcance del proyecto, permitiendo no

solo recolectar datos, sino también **interpretarlos y predecir comportamientos del tráfico urbano** en distintas condiciones.

Capítulo 3. ANALISIS DEL PROBLEMA

Necesidad y motivación

La gestión del tráfico en las grandes ciudades supone un reto constante debido a la alta densidad de vehículos y a la falta de información integrada y actualizada que permita entender los patrones de movilidad. En el caso de Madrid, el tráfico urbano genera congestión diaria, contaminación y tiempos de desplazamiento elevados, lo que evidencia la necesidad de disponer de **datos en tiempo real** que ayuden a estudiar y anticipar los comportamientos del tráfico.

A pesar de que el Ayuntamiento de Madrid ofrece diferentes fuentes abiertas como las cámaras de tráfico o los datos de incidencias, **la información se encuentra fragmentada** y con distintos formatos de publicación. Esto dificulta su uso conjunto y su análisis automatizado. Además, las fuentes meteorológicas como **Open-Meteo** o **AEMET** no están integradas con los datos de tráfico, a pesar de que el clima tiene un impacto directo en la movilidad.

En este contexto, el proyecto **TrafiVison** aborda el problema desde un enfoque práctico: **automatizar la recolección, limpieza y unificación de información visual y meteorológica**, generando un dataset real que combine ambos tipos de datos. De este modo, se busca mejorar la accesibilidad y la utilidad de los datos abiertos para el análisis académico y técnico del tráfico urbano.

Problemas detectados durante el desarrollo

Durante la primera fase del proyecto se han identificado diversos retos técnicos y de organización de los datos:

- **Formato inconsistente de la hora:** las primeras capturas incluían una fecha errónea ("1900-01-01"), lo que afectaba la coherencia temporal del dataset. Este problema se corrigió con un script de limpieza ([limpiar_hora_dataset.py](#)) que deja las horas en formato estándar HH:MM.

- **Desfase temporal entre cámaras y clima:** los datos meteorológicos no siempre coinciden exactamente con el instante de captura de las imágenes. Para corregirlo, se añadió una tolerancia de ±2 minutos en el proceso de unión.
- **Desbalance de muestras:** al haberse recogido las imágenes en tiempo real, la mayoría de ellas pertenecen a días sin lluvia. Este desbalance puede afectar al modelo de predicción y se prevé ampliarlo en futuras capturas.
- **Limitaciones de iluminación:** las imágenes nocturnas presentan reflejos y contrastes que dificultan la detección precisa de vehículos por parte de YOLOv8, por lo que se están ajustando los parámetros de confianza y los filtros de análisis.

Estos problemas se han ido resolviendo de forma progresiva, priorizando la **consistencia, la calidad del dato y la sincronización temporal** entre las distintas fuentes.

Casos de uso actuales y previstos

Actualmente, TrafiVison permite realizar un análisis inicial del comportamiento del tráfico en diferentes puntos de Madrid, combinando variables como hora, temperatura, lluvia y día de la semana.

Los principales casos de uso contemplados son:

1. **Análisis exploratorio:** estudiar cómo varía el tráfico según la franja horaria, el clima o si el día es laborable o festivo.
2. **Comparativa de zonas:** identificar qué ubicaciones presentan mayor congestión en distintos horarios o condiciones.
3. **Base para modelos predictivos:** preparar el dataset para que en el futuro pueda entrenarse un modelo capaz de anticipar el nivel de

tráfico según factores ambientales y temporales.

Requisitos del problema

- Los datos deben proceder de fuentes oficiales y mantenerse actualizados mediante un proceso automatizado.
- El sistema debe garantizar la coherencia temporal entre clima e imágenes.
- El dataset debe estar limpio, estructurado y preparado para su análisis en herramientas de aprendizaje automático o visualización.
- El proceso ETL debe ser reproducible, sin depender de intervención manual.

Perspectiva futura

En las próximas fases, el proyecto se centrará en **ampliar el volumen de datos**, incorporar nuevas cámaras y mejorar la detección nocturna.

También se prevé desarrollar un **modelo predictivo** que relacione los factores climáticos y temporales con los niveles de tráfico, así como una **interfaz visual** que permita consultar los resultados y realizar análisis dinámicos.

El objetivo es que TrafiVison evolucione desde un sistema de recolección de datos hacia una herramienta analítica completa que contribuya al estudio y mejora de la movilidad en entornos urbanos.

Análisis del problema

Necesidad y motivación

La gestión del tráfico en Madrid representa un desafío constante, especialmente en zonas con gran densidad de circulación. Las congestiones diarias, los tiempos de espera y la contaminación derivada del tráfico hacen necesario contar con **datos reales y actualizados** que permitan analizar patrones de movilidad y anticipar situaciones de alta ocupación vial.

Aunque el Ayuntamiento de Madrid ofrece acceso a múltiples fuentes abiertas, entre ellas las **cámaras de tráfico en tiempo real**, esta información suele presentarse de forma dispersa y sin integración directa con otros factores relevantes, como las condiciones meteorológicas. Esta falta de conexión entre datos dificulta realizar un análisis conjunto que permita extraer conclusiones útiles.

El proyecto **TrafiVison** se desarrolla con el propósito de **unificar y automatizar la recopilación de información visual y meteorológica**. Para ello, se han seleccionado diez cámaras ubicadas en **puntos estratégicos de la ciudad**, donde se registran flujos de tráfico representativos de diferentes tipos de vías y entornos urbanos. Las ubicaciones seleccionadas son:

- **Calle Princesa – Serrano Jover**
- **Alonso Martínez**
- **Cibeles**
- **Calle Alcalá – Velázquez**
- **Paseo de la Castellana – Santiago Delgado**
- **Avenida de América – Francisco Silvela**
- **Callao – Gran Vía**

- **Cuatro Caminos**
- **Paseo del Prado – Huertas**
- **Plaza Castilla (Norte)**

Estas calles se escogieron porque combinan **zonas céntricas, avenidas principales y ejes de conexión entre barrios**, lo que permite capturar distintos patrones de tráfico a lo largo del día. La selección también garantiza cierta diversidad en la iluminación y en las condiciones de entorno, factores clave para entrenar modelos de visión artificial.

El sistema captura imágenes de estas cámaras cada **15 minutos**, alineadas a las horas en punto y a los cuartos (:00, :15, :30, :45), y las combina con los datos climáticos obtenidos de la API **Open-Meteo**, que proporciona temperatura y precipitaciones para la ciudad de Madrid.

Problemas detectados durante el desarrollo

Durante la primera fase de implementación se identificaron y resolvieron distintos retos técnicos:

- **Formato inconsistente de hora:** inicialmente, los datos registraban la fecha "1900-01-01" junto con la hora, lo que generaba errores en la unificación de los CSV. Se resolvió con un script de limpieza que normaliza la hora a formato **HH:MM**.
- **Desfase temporal entre cámaras y clima:** la frecuencia de actualización de las cámaras y la API meteorológica no coincidían exactamente, por lo que se añadió una **tolerancia de ±2 minutos** en el proceso de unión.
- **Desbalance de datos:** la mayoría de capturas corresponden a días sin lluvia y horas diurnas, lo que produce una representación desigual. Se planea continuar la recolección para equilibrar las condiciones

meteorológicas y temporales.

- **Detección nocturna limitada:** las condiciones de baja iluminación afectan la precisión de YOLOv8, especialmente en zonas con farolas o reflejos. Actualmente se están evaluando ajustes de parámetros y filtros adicionales.

Estos problemas se abordaron de forma progresiva, con el objetivo de asegurar la **coherencia temporal y la calidad de los datos**, priorizando la fiabilidad del dataset frente al volumen de información recolectada.

Casos de uso actuales y previstos

En su estado actual, TrafiVison permite realizar un análisis inicial de los niveles de tráfico en distintos puntos de Madrid. Entre los principales usos se incluyen:

1. **Análisis exploratorio:** estudiar cómo varía la intensidad del tráfico según la hora del día, la ubicación y las condiciones meteorológicas.
2. **Comparativa entre zonas:** observar diferencias de ocupación entre calles céntricas (por ejemplo, Gran Vía o Cibeles) y zonas de conexión (como Plaza Castilla o Cuatro Caminos).
3. **Base para predicción futura:** preparar un dataset que permita entrenar modelos de aprendizaje automático capaces de anticipar niveles de tráfico según clima y hora.

Requisitos del problema

- Los datos deben ser reales, abiertos y obtenidos automáticamente de las fuentes oficiales.

- El proceso ETL debe funcionar de forma periódica y sin intervención manual.
- La hora y la fecha deben mantenerse sincronizadas entre cámaras y clima.
- El dataset final debe ser limpio, coherente y adecuado para análisis estadístico o predictivo.

Perspectiva futura

En las próximas fases se prevé ampliar la cantidad de imágenes recogidas y **prolongar el periodo de observación** para incluir más condiciones meteorológicas (lluvia, nubes, días soleados).

También se planea desarrollar un **modelo predictivo** que utilice las variables de hora, temperatura, lluvia y tipo de día (laborable o festivo) para estimar el nivel de tráfico.

Asimismo, se contempla integrar un **módulo de visualización** que muestre mapas de congestión o gráficos comparativos por zonas y franjas horarias.

De este modo, TrafiVison busca evolucionar desde un sistema de recolección de datos hacia una herramienta completa de análisis y predicción del tráfico urbano en Madrid.

Capítulo 4. PROPUESTA DE SOLUCIÓN (DISEÑO)

En este capítulo se debe incluir una explicación de:

Como se va a abordar la solución del problema. Se explican elementos como la arquitectura software / hardware que se van a usar, como se divide el problema, elementos tecnológicos (servidores, aplicaciones, etc etc)

4.1 Organización de las subtareas

El diseño del proyecto **TrafiVision** se centra en la creación de una arquitectura modular capaz de **automatizar la captura, limpieza y análisis de datos de tráfico** en la ciudad de Madrid.

La solución propuesta se basa en un proceso **ETL (Extracción, Transformación y Carga)** completamente automatizado, al que posteriormente se le añadirá una capa de **modelado predictivo y visualización interactiva**.

El sistema se divide en tres grandes módulos:

1. **Extracción automática de datos (E)** — obtención de imágenes y datos meteorológicos.
2. **Transformación y limpieza (T)** — depuración, clasificación y unificación en un único dataset.
3. **Carga y análisis (L)** — almacenamiento del dataset final y preparación para el modelado predictivo.

Cada módulo se implementa mediante scripts en **Python**, integrando librerías como **requests**, **pandas**, **schedule**, **ultralytics** y **os**, junto con un sistema de carpetas organizado para almacenar tanto las imágenes como los CSV generados.

Arquitectura general

El proyecto se ejecuta sobre una arquitectura **software totalmente local**, organizada en tres niveles:

- **Nivel de datos:** recopilación automática de información desde las fuentes abiertas de la DGT y Open-Meteo.
- **Nivel de procesamiento:** scripts de Python que limpian, integran y transforman los datos.
- **Nivel de análisis:** uso de herramientas de IA (AI Studio / Alair) para generar modelos predictivos de tráfico.

No se requiere hardware adicional más allá del equipo de desarrollo, aunque el diseño es escalable para su ejecución en servidores o plataformas en la nube (por ejemplo, Google Colab o Azure Notebooks).

Organización de subareas

Subtarea 1: Selección y análisis de fuentes de datos

- **Descripción:**

Se seleccionaron fuentes de información públicas y actualizables en tiempo real, priorizando su fiabilidad y formato estandarizado.
Las fuentes elegidas fueron:

 - Cámaras de tráfico del Ayuntamiento de Madrid (imágenes en formato JPG actualizadas periódicamente).
 - API de Open-Meteo (datos JSON con temperatura, precipitaciones y códigos de clima).
- **Recursos:**
 - Librerías: `requests`, `os`, `datetime`.

- Formatos de archivo: `.jpg` (imágenes) y `.json` (datos meteorológicos).
- **Motivo de elección:**

Ambas fuentes son públicas, accesibles y presentan una estructura regular que facilita la automatización.
- Subtarea 2: Descarga y almacenamiento de imágenes**
 - **Descripción:**

A través del script `etl_camaras_madrid.py`, se implementó un proceso automatizado que descarga imágenes de **10 cámaras clave de Madrid** (Alcalá, Cibeles, Plaza Castilla, Gran Vía, etc.) cada 15 minutos. Cada imagen se guarda con su nombre, fecha y hora en la carpeta `imagenesTrafico`.
 - **Entrada:** URLs de las cámaras seleccionadas.
 - **Salida:** Imágenes locales con nombre estructurado (`calle_fecha_hora.jpg`).
 - **Recursos:**
 - Librerías: `requests`, `os`, `datetime`.
 - Carpeta de destino: `imagenesTrafico`.
 - **Comentarios:**

Durante las primeras pruebas se detectaron errores de conexión en algunas cámaras, lo que permitió ajustar los tiempos de descarga para evitar interrupciones.

Subtarea 3: Obtención de datos meteorológicos

- **Descripción:**

Se desarrolló el script `etl_tiempo.py`, encargado de consultar la API de

Open-Meteo para obtener temperatura y precipitaciones en tiempo real.

El script se sincroniza con el proceso de las cámaras, ejecutándose también cada 15 minutos y guardando la información en [tiempo_madrid.csv](#).

- **Entrada:** Coordenadas geográficas de Madrid (40.4168, -3.7038).
 - **Salida:** CSV con columnas [fecha](#), [hora](#), [temperatura](#), [lluvia](#), [litros_m2](#).
 - **Recursos:**
 - Librerías: [requests](#), [pandas](#), [schedule](#).
 - Carpeta destino: [csv](#).
 - **Incidencias:**

Inicialmente, la hora se registraba con formato erróneo (añadía una fecha fantasma), problema que se resolvió en la fase de depuración posterior.
- ### Subtarea 4: Análisis de imágenes mediante YOLOv8
- **Descripción:**

En esta fase se aplicó el modelo **YOLOv8** (de la librería [ultralytics](#)) para analizar cada imagen descargada y detectar vehículos. El script [yolo_final.py](#) ejecuta el modelo sobre todas las imágenes y genera un CSV ([yolo_final.csv](#)) con el número de vehículos detectados y el nivel de tráfico estimado (bajo, medio o alto).
 - **Entrada:** Carpeta [imagenesTrafico](#).
 - **Salida:** CSV con columnas [foto](#), [num_vehiculos](#), [nivel_trafico](#).
 - **Recursos:**

- Librerías: `ultralytics`, `pandas`, `os`.
- Modelo: `yolov8n.pt` (versión ligera para uso local).

- **Resultados:**

El modelo permitió generar etiquetas automáticas para más de 2.900 imágenes, lo que constituye la base del dataset final.

Subtarea 5: Limpieza, integración y generación del dataset final

- **Descripción:**

Se unificaron los tres archivos (`camaras_solo.csv`, `tiempo_madrid.csv` y `yolo_final.csv`) en un único dataset mediante el script `merge_datasets_final.py`.

Este proceso incluyó la limpieza de columnas, corrección de la hora y creación de nuevas variables derivadas como:

- Día laborable / no laborable.
- Franja horaria (madrugada, mañana, mediodía, tarde, noche).

- **Entrada:** Archivos CSV generados por las etapas anteriores.

- **Salida:** `dataset_final_limpio.csv`.

- **Recursos:**

- Librerías: `pandas`, `datetime`, `os`.
- Carpeta destino: `csv`.

- **Comentarios:**

Durante la integración se detectaron discrepancias de pocos minutos entre el CSV de cámaras y el del clima, lo cual se resolvió ajustando las coincidencias por rango horario.

Subtarea 6: Preparación para el modelado

- **Descripción:**

Con el dataset limpio y estructurado, se deja preparada la base para la fase de **entrenamiento del modelo predictivo**.

Esta etapa incluirá la conexión del CSV con una herramienta de IA visual como **AI Studio o Alair**, donde se probarán distintos algoritmos de clasificación.

- **Recursos futuros:**

- Plataformas: AI Studio / Azure ML.
- Entrada: [dataset_final_limpio.csv](#).
- Salida esperada: modelo predictivo capaz de estimar el nivel de tráfico.

Subtarea 7: Versionado y documentación

Descripción:

Finalmente, el proyecto completo ha sido subido a un repositorio de GitHub para su control de versiones, permitiendo un trabajo colaborativo entre ambos integrantes (Rodrigo y Anyi).

Este paso asegura trazabilidad, copias de seguridad y control del avance.

Recursos:

4.2 Herramienta: Git y GitHub.

4.3 Archivos incluidos: scripts ETL, dataset, modelos y documentación.

Capítulo 5. DESARROLLO / IMPLEMENTACIÓN

Dado que se ha detectado el problema, se ha propuesto una solución y un diseño de la solución, ahora hay que describir la implementación.

La implementación puede ocupar varios capítulos en función del proyecto.

Por ejemplo la propuesta económica debería estar al final de la parte de implementación

Capítulo 6. Modelado

Modelado

Estado actual del modelado

En la fase actual del proyecto **TrafiVision**, el trabajo se ha centrado en la **preparación y depuración del dataset base**, que servirá como punto de partida para el entrenamiento de los modelos de predicción.

A día de hoy, el equipo ha completado la **construcción automatizada del conjunto de datos**, integrando tres fuentes principales:

1. **Imágenes de tráfico** obtenidas en tiempo real desde diez cámaras del Ayuntamiento de Madrid.
2. **Datos meteorológicos** proporcionados por la API de Open-Meteo, con variables como temperatura y precipitación.
3. **Etiquetas de tráfico** generadas automáticamente mediante el modelo YOLOv8, que clasifica las imágenes según el número de vehículos detectados (tráfico bajo, medio o alto).

El resultado de esta integración es el archivo **dataset_final_limpio.csv**, un conjunto de datos estructurado y listo para análisis.

Cada registro combina información visual, temporal y climática, junto con variables derivadas como:

- Día laborable / no laborable
- Franja horaria (madrugada, mañana, mediodía, tarde o noche)
- Nivel de tráfico estimado

Este dataset se considera **la base fundamental** sobre la que se construirá la fase predictiva del proyecto.

Objetivo del modelado

El objetivo principal del modelado en TrafiVision consiste en **predecir el nivel de tráfico** a partir de variables meteorológicas, temporales y contextuales.

El modelo deberá ser capaz de **identificar patrones de congestión** y anticipar la intensidad del tráfico en diferentes zonas y horarios de Madrid, permitiendo en el futuro generar alertas o recomendaciones.

Herramientas de modelado

Para la creación y evaluación del modelo se emplearán plataformas de inteligencia artificial accesibles y visuales, como:

- **AI Studio (Google o IBM Watson)**, para pruebas rápidas y conexión directa con CSV.
- **Azure Machine Learning Studio (Alair)**, que ofrece flujos de trabajo visuales para entrenar, validar y desplegar modelos sin necesidad de código adicional.

Estas herramientas permitirán conectar el dataset generado localmente o desde GitHub, probar diferentes algoritmos y comparar resultados de manera estructurada.

Fases del proceso de modelado

1. Preparación del dataset

- Verificación del formato y consistencia del archivo `dataset_final_limpio.csv`.
- Carga del dataset en la plataforma elegida (AI Studio o Alair).

- División del dataset en **80% entrenamiento** y **20% prueba**, manteniendo equilibrio entre clases (bajo, medio, alto).

2. Selección de variables

Se utilizarán como variables de entrada:

- **Temperatura**
- **Lluvia (en litros/m²)**
- **Hora del día**
- **Tipo de día** (laborable / no laborable)
- **Franja horaria**

Y como **variable objetivo**, el campo **nivel_tráfico**.

3. Entrenamiento de modelos

Se probarán varios algoritmos de clasificación supervisada, entre ellos:

- **Regresión logística**, para un modelo lineal básico.
- **Árboles de decisión y Random Forest**, para capturar relaciones no lineales.
- **k-Nearest Neighbors (kNN) y Support Vector Machines (SVM)**, para comparar enfoques.
- En fases futuras, se evaluará el uso de **redes neuronales simples** si el tamaño del dataset crece lo suficiente.

Cada modelo será evaluado con métricas de precisión, recall, F1-score y matriz de confusión, para determinar su rendimiento.

4. Evaluación y validación

El modelo más eficiente será seleccionado en función de su equilibrio entre exactitud y capacidad de generalización.

Se generarán informes automáticos de rendimiento y se documentarán los resultados para su inclusión en la memoria final.

Resultados esperados

A corto plazo, se espera que el modelo logre identificar correctamente el **nivel de tráfico en función de la hora y el clima** con una precisión mínima del 75%.

A medio plazo, se busca ampliar el dataset para incluir **nuevas variables contextuales**, como velocidad media estimada o densidad de vehículos por zona.

Una vez validado, el modelo se integrará dentro del pipeline de TrafiVision, de modo que **cada nuevo lote de datos procesados (imágenes + clima)** pueda alimentar automáticamente la predicción.

Perspectiva futura

En fases posteriores, el proyecto incorporará un **módulo de visualización interactiva** (por ejemplo, con Streamlit o Power BI), donde se representarán los resultados del modelo de forma gráfica.

Se podrán observar mapas de calor por zonas, curvas de congestión horaria o comparativas de tráfico según temperatura o lluvia.

Además, el sistema permitirá **entrenamientos periódicos automáticos**, de manera que el modelo se actualice a medida que se recojan nuevos datos.

En resumen, la fase de modelado representa el paso hacia la transformación de TrafiVision **de un sistema de recopilación de datos** a una **herramienta predictiva inteligente**, capaz de anticipar y analizar patrones reales de movilidad urbana en Madrid.

Capítulo 7. BIBLIOGRAFIA/WEBGRAFIA

Fuentes técnicas y documentación

- [1] Open-Meteo. API Documentation: Free Weather Forecast API for Developers. Disponible en: <https://open-meteo.com/>
Consultado en octubre de 2025.
- [2] Ayuntamiento de Madrid. Portal de Datos Abiertos: Tráfico en tiempo real y cámaras urbanas. Disponible en:
<https://informo.madrid.es/informo/tmadrid/camaras.jsp>
Consultado en octubre de 2025.
- [3] Ultralytics. YOLOv8 Documentation — Object Detection Models. Disponible en:
<https://docs.ultralytics.com/>
Consultado en noviembre de 2025.
- [4] Python Software Foundation. Python 3.9 Documentation. Disponible en:
<https://docs.python.org/3.9/>
- [5] Pandas Development Team. Pandas User Guide — Data Analysis Library. Disponible en:
<https://pandas.pydata.org/docs/>
- [6] Schedule Library. Python Job Scheduling for Humans. Disponible en:
<https://schedule.readthedocs.io/>
- [7] Requests Library. Python HTTP for Humans. Disponible en:
<https://requests.readthedocs.io/>
- [8] Azure Machine Learning Studio (Alair). Documentación de Azure Machine Learning. Disponible en:
<https://learn.microsoft.com/es-es/azure/machine-learning/>
- [9] Google AI Studio. Herramienta de experimentación en IA. Disponible en:
<https://aistudio.google.com/>

Fuentes académicas y metodológicas

[10] AENOR. AEN/CTN 157 — Proyectos. Normas y Publicaciones. [En línea] 2010. Disponible en:

<http://www.aenor.es/aenor/normas/ctn/fichactn.asp?codigonorm=AEN/CTN%20157>

[11] UNE 157001. Criterios generales para la elaboración de proyectos. Escuela Universitaria de Ingeniería de Vitoria. 2002.

[12] Universidad Europea de Madrid. Guía de estilo y normativa para la redacción de proyectos de ingeniería.

Disponible en:

<http://biblioteca.uem.es/es/aprendizaje-y-formacion/citas-bibliograficas-documentos>

Capítulo 8. ANEXOS

ANEXO I – Criterios de clasificación y variables derivadas

Durante la fase de diseño del dataset, se establecieron una serie de criterios lógicos para convertir los datos numéricos y contextuales en variables categóricas útiles para el análisis y el futuro modelado predictivo.

Estas reglas fueron implementadas directamente dentro de los scripts de transformación y limpieza, y se aplican de forma automática a cada nuevo registro obtenido por el sistema ETL.

1. Clasificación del nivel de tráfico

El nivel de tráfico se determinó en función del **número de vehículos detectados por el modelo YOLOv8** en cada imagen capturada.

Para convertir este valor numérico en una categoría descriptiva, se establecieron los siguientes rangos:

Nº de vehículos detectados	Nivel de tráfico asignado
0 – 5	Bajo
6 – 15	Medio
> 15	Alto

Estos umbrales se definieron tras observar los resultados iniciales del modelo y su correspondencia visual con las imágenes de las cámaras.

El criterio busca un equilibrio entre la densidad visual y la legibilidad analítica del dataset, facilitando su uso en modelos de clasificación supervisada.

2. Clasificación del tipo de lluvia

La variable “lluvia” procede del valor de **precipitación en litros/m²** proporcionado por la API de Open-Meteo.

Dado que pequeñas cantidades no representan necesariamente lluvia perceptible, se definieron los siguientes intervalos para su categorización:

Precipitación (L/m ²)	Clasificación
≤ 0.1	No llueve
0.1 – 0.3	Lluvia débil
> 0.3	Lluvia intensa

Estos valores permiten distinguir entre condiciones secas, lloviznas leves y precipitaciones significativas, las cuales influyen directamente en la variación del tráfico urbano.

3. Categorización por franja horaria

Cada registro del dataset incluye la hora exacta de captura, la cual se transformó en una **franja horaria** representativa del momento del día.

Esta variable es relevante porque el flujo de tráfico varía de forma predecible según la hora.

Las franjas establecidas fueron:

Franja horaria	Rango horario (hh:mm)	Descripción
Madrugada	05:00 – 07:59	Tráfico bajo, desplazamientos esenciales.

Mañana	08:00 – 12:59	Horas punta laborales, mayor densidad de tráfico.
Mediodía	13:00 – 16:59	Actividad intermedia, pausas laborales y comidas.
Tarde	17:00 – 20:59	Segunda hora punta de retorno a casa.
Noche	21:00 – 04:59	Menor circulación y tráfico reducido.

Este formato permite que los algoritmos de predicción puedan relacionar patrones de tráfico con momentos específicos del día, manteniendo la coherencia temporal en todos los registros.

4. Clasificación de días laborables y no laborables

Para mejorar la capacidad predictiva del modelo, se añadió una variable binaria que distingue entre **días laborables y no laborables**, basada en el calendario oficial de la Comunidad de Madrid.

Tipo de día	Criterionio
Laborable	Lunes a viernes (excepto festivos).
No laborable	Sábados, domingos y festivos oficiales.

Esta variable permite detectar diferencias de comportamiento entre el tráfico diario y el de fines de semana o días festivos, que suelen mostrar patrones menos congestionados.

Resumen

Los criterios anteriores aseguran que el dataset generado por TrafiVision no solo contenga valores numéricos, sino también **información semántica interpretable**, lo que facilita su análisis exploratorio y el entrenamiento de modelos de aprendizaje automático con variables claras, coherentes y fácilmente escalables.