

SEMÁFORO INTELIGENTE AUTÓNOMO

TRABAJO DE FIN DE MÁSTER

MASTER IN BIG DATA AND BUSINESS INTELLIGENCE

Esmeralda Arribas Calderón

Gisela María Gpe Bernal Avila

Judeh Samir Saba Ibarra

Director del Máster: Belarmino García

Tutor del TFM: Miguel Couto

Madrid, a 09 de Junio de 2025

Abstract

El crecimiento acelerado de la población urbana y el aumento de la congestión vehicular plantean importantes desafíos para la movilidad en las ciudades. Los sistemas de semáforos tradicionales, con tiempos fijos o controlados de forma reactiva, a menudo no logran optimizar el flujo del tráfico, lo que genera mayores tiempos de viaje, incremento en el consumo de combustible y un aumento en las emisiones de CO₂. Adicionalmente, recientemente junto con la falta de electricidad en la totalidad de la península ibérica, en este contexto, los semáforos inteligentes autónomos, impulsados por Inteligencia Artificial (IA) y Big Data, surgen como una solución prometedora para mejorar la eficiencia del tráfico, la seguridad vial y la sostenibilidad en entornos urbanos.

Este estudio realiza una revisión bibliográfica y un análisis de factibilidad para el desarrollo de un sistema autónomo de semáforos basado en IA. La investigación identifica tecnologías clave (aprendizaje automático, visión por computadora, edge computing), bases de datos públicas de tráfico. Asimismo, examina los desafíos técnicos, legales y éticos, como la fiabilidad de los sensores, la privacidad de los datos y la equidad en la priorización del tráfico.

Se desarrolla un modelo de simulación en Python y PyGame para probar ajustes dinámicos en los semáforos según el flujo vehicular y peatonal en tiempo real. El sistema integra sensores basados en Raspberry Pi, detección de objetos con YOLO y datos abiertos de tráfico de Madrid para optimizar los tiempos de las señales.

El sistema propuesto en este trabajo, demuestra viabilidad técnica. El estudio concluye que los semáforos inteligentes pueden mejorar significativamente la movilidad urbana, sin embargo requieren mayor investigación.

INDICE

Abstract	2
INDICE	3
1. Introducción	5
1.1. Antecedentes - Bases de Investigación	5
1.2. Justificación	10
1.3. Problema, Hipótesis y Objetivos	12
1.4. Metodología	13
2. Marco teórico	17
2.1. Tecnologías de Gestión de Bases de Datos	17
2.2. Machine Learning	19
2.3. Visión artificial	20
3. Cuerpo central del Trabajo de campo/Desarrollo	21
3.1. Descripción del Flujo del Sistema de Monitoreo de Tráfico	23
3.2. Software basado en bases de datos del Ayuntamiento de Madrid	24
3.2.1. Arquitectura de SQL Azure	25
3.2.2. Análisis de Tráfico Vehicular	27
3.2.3. BBDD tráfico peatonal	33
3.2.4. BBDD bicicletas	36
3.2.5. BBDD accidentalidad	38
Modelo predictivo	43
3.3. Software basado en visión computacional	45
3.3.1. Obtención de imágenes reales (Web scraping)	45
3.3.2. Detección de vehículos y coches en imágenes	46
3.3.3. Código de simulación	47
3.4. Hardware	50
3.4.1. Elementos para el prototipo	51
3.4.2. Análisis de autonomía operacional	52
3.5. Problemas Técnicos y Tecnológicos	55
4. Análisis y resultados	60
4.1. Sistema basado en modelo predictivo	61
Resultados Obtenidos	61
4.2. Sistema basado en visión computacional	64
4.3. Análisis de costes	67
4.4. Análisis de reducción de huella de carbono	68
5. Propuesta de líneas futuras de investigación	69
6. Conclusiones	70
Hardware	76

7. Bibliografía

1. Introducción

En este capítulo introductorio se realiza un estudio de semáforos inteligentes autónomos, se analizan las oportunidades y desafíos para el proyecto, se plantean una serie de objetivos y una metodología matriz a seguir en su desarrollo.

Como se especifica en el Abstract, el problema de semáforos con tiempos fijos, provoca viajes con tiempos más largos, mayor consumo de combustible y un aumento de las emisiones de CO₂. En este contexto, los semáforos inteligentes que utilizan inteligencia artificial (IA) y big data prometen la eficiencia del tráfico, la seguridad vial y la sostenibilidad en entornos urbanos. El mercado de la IA en el transporte se estima que alcanzará los 4,11 billones de dólares en 2024, lo que demuestra la creciente importancia económica de este campo.

Este trabajo presenta un estudio de la literatura y las bases de datos relevantes para el desarrollo de un modelo de semáforos inteligentes autónomos. El objetivo principal es identificar las tecnologías, los desafíos y las oportunidades de investigación en este campo, con el fin de crear un sistema que se gestione por sí mismo de manera eficiente y autónoma.

1.1. Antecedentes - Bases de Investigación

Semáforos Inteligentes e IA en la Gestión del Tráfico

La investigación en semáforos inteligentes se ha centrado en el desarrollo de sistemas que se adaptan dinámicamente a las condiciones del tráfico en tiempo real. Diversos enfoques, como la lógica difusa, el aprendizaje automático y las redes neuronales.

Estas soluciones garantizan el poder manejar grandes conjuntos de datos, que abarcan patrones de tráfico históricos, condiciones climáticas y eventos especiales

como accidentes, para predecir escenarios de tráfico futuros. Esta capacidad predictiva permite a las autoridades de transporte implementar medidas proactivas, como ajustar los tiempos de las señales o desplegar recursos adicionales, para optimizar los tiempos de los semáforos y mejorar el flujo del tráfico.

Se están llevando a cabo varios proyectos de investigación de la aplicación de la IA en la gestión del tráfico en universidades y empresas, entre los cuales destacan los siguientes:

Surtrac: Proyecto de semáforos inteligentes de la Universidad Carnegie Mellon (Pittsburg, USA) [1]: Este proyecto piloto ha demostrado que los semáforos inteligentes pueden reducir significativamente las emisiones de los vehículos y los tiempos de viaje. El proyecto se centra en una tecnología de control de señales de tráfico adaptativo que permite que el tráfico se comuniquen entre sí y se adapten de forma colaborativa a las condiciones reales del tráfico en tiempo real.

City Brain: Sistema de gestión de tráfico de IA desarrollado por Alibaba (Hangzhou, China) [2]: El sistema utiliza el aprendizaje automático y la visión artificial para monitorear las condiciones del tráfico y gestionar el control de las señales. Con el sistema los vehículos pueden circular a velocidades algo más elevadas (+15%). Además, en Hangzhou se ha mejorado la precisión en la detección de incidentes en un 92%. Esto ha hecho que los tiempos de respuesta de los servicios de emergencia se reduzca notablemente en unos 3 minutos de media y un 50% de probabilidades de llegar en 7 minutos tras el accidente.

RHODESNG con IntelliDriveSM: Proyecto basado en Modelos de IA [3]: Este proyecto de la Administración Federal de Carreteras de EE. UU. se centra en el

desarrollo de una arquitectura y algoritmos de próxima generación para sistemas de control de tráfico que se adaptan continuamente a las condiciones cambiantes del tráfico a través de comunicaciones de alta velocidad con vehículos e infraestructura. El sistema utiliza algoritmos autoadaptativos para integrar los datos de posición, velocidad y cola recibidos de los vehículos y los sensores y transmisores de la infraestructura, realizar cálculos precisos de alta velocidad, hacer predicciones y ajustar continuamente sus parámetros críticos en función de los datos entrantes.

HumanLight [4]: Este proyecto de la Universidad de California en Berkeley utiliza la IA para priorizar los vehículos de alta ocupación en las intersecciones. El objetivo es optimizar el control de las señales de tráfico para maximizar el rendimiento de las personas, en lugar del rendimiento de los vehículos. HumanLight utiliza una técnica de IA, el aprendizaje por refuerzo, para gestionar el comportamiento dinámico de entornos de tráfico complejos.

Proyecto KI4LSA y KI4PED del Instituto IOSB [5]: Este proyecto utiliza IA para permitir el control inteligente de los semáforos con el fin de reducir los atascos de tráfico y mejorar la seguridad de los peatones en los cruces. El proyecto utiliza cámaras de alta resolución y sensores de radar para captar la situación real del tráfico con mayor precisión, lo que permite determinar con exactitud el número de vehículos que esperan en un cruce en tiempo real.

Una vez habiendo analizado diferentes tipologías de sistemas y la viabilidad de las mejoras implementadas, se realiza un estudio de fuentes de datos públicas con distintos formatos

Bases de Datos Públicas

Las bases de datos encontradas han sido clasificadas en tres categorías: información de tráfico en tiempo real, patrones de movilidad, accidentes y condiciones climáticas.

1. Información de Tráfico en Tiempo Real

Existen varias bases de datos públicas que proporcionan información sobre el tráfico en tiempo real. Las más relevantes se muestran en la Tabla 1.

Base de Datos	Proveedor	Características
HERE Real-Time Traffic	HERE Technologies	Datos de flujo (actualizados cada minuto) e incidentes, cobertura global
StreetLight Data	StreetLight Data, Inc.	Volúmenes y velocidades a la carta, datos históricos, análisis de rutas
TomTom Traffic Stats	TomTom	Base de datos histórica de tráfico, análisis de la red de carreteras, informes en tiempo real
Mapbox Traffic Data	Mapbox	Tráfico en vivo y típico, cobertura global, algoritmos de aprendizaje automático
ArcGIS Online World Traffic Service	Esri	Datos de tráfico en vivo y predictivos, integración con ArcGIS StreetMap Premium

Tabla 1: Tipos de bases de datos de tráfico en tiempo real

2. Patrones de Movilidad

Las bases de datos, presentadas en la Tabla 2 ofrecen información valiosa sobre los patrones de movilidad.

Base de Datos	Proveedor	Características
Mobility in Cities Database	UITP	Indicadores de transporte para más de 60 áreas metropolitanas
Mobility Data	DataRade	Datos de tráfico peatonal, GPS, transporte público y aplicaciones de movilidad
Human Mobility Datasets	CARTO	Datos de movilidad humana anonimizados y agregados
Mobility Data	SafeGraph	Datos de movilidad global, análisis de áreas comerciales, selección de sitios minoristas
The Mobility Database	MobilityData	Catálogo internacional de datos de transporte público

Tabla 2: Tipos de bases de datos de patrones de movilidad

3. Accidentes y Condiciones Climáticas

Para obtener información sobre accidentes y condiciones climáticas, se pueden consultar las bases de datos recogidas en la Tabla 3.

Base de Datos	Proveedor	Características
Storm Events Database	NCEI, NOAA	Registros de tormentas y fenómenos meteorológicos significativos

EM-DAT	CRED	Datos sobre la ocurrencia e impactos de desastres a nivel mundial
NTSB Aviation Accident Database	NTSB	Base de datos de accidentes de aviación civil

Tabla 3: Tipos de bases de datos de accidentes y clima.

Teniendo en cuenta los casos de éxito previamente descritos, se plantea que el sistema para el control inteligente de semáforos debe combinar elementos de software y de hardware. Por un lado, la extracción de datos reales mediante elementos como sensores o cámaras. Por otro, el preprocesamiento de los datos, el procesamiento de estos y el algoritmo para predecir las condiciones del tráfico y los tiempos de los semáforos mediante algoritmos de aprendizaje automático. Por ejemplo mediante el empleo de random forest. Se propone que el programa siga una lógica de escalera donde clasifica el tráfico en volúmenes bajos, medios, altos y muy altos, y en base a esto, se modifica el flujo de tráfico.

1.2. Justificación

Los sistemas tradicionales, los cuales suelen operar con temporizadores fijos o sensores básicos sin capacidad de adaptación en tiempo real, generan congestión al no responder dinámicamente a las fluctuaciones del tráfico. Mientras que las nuevas tecnologías vienen para cambiar el paradigma actual. de un sistema de control inteligente de semáforos basado en tecnologías modernas de Inteligencia Artificial (IA) y Big Data permite una gestión dinámica capaz de responder a incidentes en tiempo real, patrones estacionales e incluso eventos imprevistos, maximizando la eficiencia de la red vial.

Además de la reducción de los atascos, también disminuyen el tiempo de viaje, el consumo de combustible y las emisiones de CO₂, contribuyendo así a una ciudad más sostenible y respetuosa con el medio ambiente y reforzando el compromiso con los Objetivos de Desarrollo Sostenible (ODS) [6].

Actualmente, las ciudades líderes en movilidad inteligente, como Singapur, algunas metrópolis de EE.UU. (e.g., Los Ángeles, Pittsburgh) y China (e.g., Hangzhou, Shenzhen), ya han implementado sistemas de semáforos inteligentes con resultados demostrables: reducciones de hasta un 30% en los tiempos de desplazamiento y un 20% en emisiones contaminantes. Se selecciona Madrid, como capital innovadora, con potencial de posicionarse a la vanguardia de las smart cities europeas adoptando estas tecnologías, en línea con los objetivos de la Agenda 2030 y el Pacto Verde Europeo.

Este proyecto no solo mejoraría la calidad de vida de los ciudadanos, sino que también reforzaría el compromiso de Madrid con la digitalización y la sostenibilidad, transformando su infraestructura urbana en un referente global de eficiencia y reducción de la huella de carbono. La inversión en IA para la gestión del tráfico no es solo una apuesta por la innovación, sino una necesidad en un mundo donde la movilidad eficiente y ecológica es clave para el futuro de las ciudades.

En un contexto global donde la competencia entre ciudades por atraer talento e inversión se centra en la digitalización y la sostenibilidad, Madrid no puede permitirse quedarse atrás. Implementar un sistema de semáforos inteligentes no solo mejoraría la movilidad, sino que posicionaría a la capital española como líder en la transformación hacia la smart city [7], siguiendo los ejemplos donde estas tecnologías ya han demostrado su valor económico y ecológico.

1.3. Problema, Hipótesis y Objetivos

El objetivo principal es contribuir al desarrollo de un sistema de semáforos inteligentes autónomo que sea seguro, eficiente, sostenible y ético, y que pueda mejorar la movilidad urbana en las ciudades del futuro. En relación a esto, se desarrollan objetivos específicos, los cuales se describen a continuación.

El desarrollo de semáforos inteligentes surge como respuesta a los graves problemas de **congestión vehicular y peatonal** en las ciudades modernas, donde los sistemas tradicionales de control de tráfico resultan inflexibles e inefficientes para manejar el flujo dinámico, lo que motiva la creación de un sistema adaptativo que **optimice la movilidad** urbana mediante IA y sensores en tiempo real.

La **dependencia** de los sistemas actuales de infraestructuras energéticas centralizadas, que los hace vulnerables ante fallos o desastres naturales, impulsa el objetivo de diseñar **semáforos autosuficientes** con paneles solares y baterías de respaldo, garantizando operatividad continua incluso en situaciones adversas.

Las **altas emisiones** de carbono generadas por el tráfico detenido y los motores en ralentí obligan a implementar una solución que reduzca activamente la **contaminación** mediante una gestión más fluida del tránsito, alineándose con los **Objetivos de Desarrollo Sostenible** (ODS) de ciudades sostenibles y energía limpia.

La **limitación** de los sistemas de tráfico convencionales, diseñados para ubicaciones específicas sin capacidad de **adaptación a otras ciudades**, plantea la necesidad de desarrollar un modelo escalable basado en parámetros ajustables y

arquitecturas de IA transferibles, facilitando su replicación en distintas zonas geográficas.

La subutilización de los **grandes volúmenes de datos** recopilados por sensores urbanos, que no son aprovechados para una toma de decisiones inteligente, justifica la integración de **técnicas de big data** y análisis predictivo que permitan identificar patrones de congestión y optimizar rutas de forma proactiva.

Finalmente, la opacidad de muchos sistemas de IA, que operan como "**cajas negras**" sin posibilidad de auditar sus decisiones o detectar sesgos, exige que este proyecto priorice la **transparencia** mediante herramientas de explicabilidad y métricas de equidad, asegurando que los beneficios del sistema sean distribuidos de forma justa entre todos los distritos.

1.4. Metodología

La metodología integra tres pilares fundamentales: **Data Science** para el modelado predictivo, **Data Analytics** para el procesamiento de información e **Ingeniería de Datos** para la arquitectura técnica, utilizando un conjunto robusto de herramientas que incluye Excel para manipulación básica, SQL para gestión de bases de datos, y Python con sus librerías especializadas (Pandas, Matplotlib, Seaborn) para análisis avanzado y visualización.

El proceso de análisis del proyecto sigue una metodología estructurada basada en el ciclo estándar de Data Analytics [8], diseñado para transformar datos en insights accionables. En la fase inicial **ASK (Preguntar)**, se enfoca en definir con precisión los objetivos estratégicos y formular las preguntas clave que guiarán toda la

investigación, tales como "¿Cómo optimizar los tiempos de los semáforos para reducir las congestiones vehiculares en horas pico?" o "¿De qué manera se puede minimizar las emisiones de CO₂ mediante una gestión inteligente del tráfico?". Este planteamiento fundamental establece el marco de referencia para todas las etapas posteriores.

La etapa **PREPARE (Preparar)** implica un trabajo meticoloso de recolección y evaluación de datos. El proyecto se basa en fuentes oficiales como el Ayuntamiento de Madrid para obtener datos de tráfico en tiempo real, y en la AEMET para información meteorológica detallada. Este proceso incluye una inspección minuciosa de la estructura de los datos, verificación de su integridad, y un filtrado riguroso para seleccionar únicamente la información relevante para los objetivos marcados anteriormente. La preparación de los datos contempla el diseño de una estructura de base de datos relacional en SQL Server que integra múltiples tablas interconectadas, permitiendo **cruces analíticos** sofisticados entre variables de tráfico, climatológicas y de accidentabilidad.

El core del proceso reside en la fase **PROCESS (Procesar)**, donde se realiza una transformación profunda de los datos crudos. Esto incluye operaciones de limpieza para manejar valores faltantes o inconsistentes, normalización de formatos, y creación de variables derivadas mediante agregaciones estadísticas (cálculo de promedios móviles, identificación de valores máximos y mínimos por franja horaria, entre otros). Se implementan técnicas de enriquecimiento de datos para combinar múltiples fuentes y generar un conjunto unificado listo para el análisis.

En la fase **ANALYZE (Analizar)**, se aplica un enfoque multidimensional que combina cuatro tipos de análisis complementarios. El **análisis descriptivo**

proporciona una comprensión básica de las tendencias históricas, mientras que el **diagnóstico** profundiza en las relaciones causales subyacentes. La componente **predictiva**, apoyada en técnicas de inteligencia artificial, permite anticipar escenarios futuros, y el **análisis prescriptivo** deriva recomendaciones concretas para la optimización de los tiempos semafóricos. La visualización de resultados se realiza principalmente mediante librerías de Python, generando tanto gráficos estáticos como representaciones dinámicas que facilitan la comparación entre la Gran Vía y otras zonas de la ciudad, incluyendo mapas de calor para accidentes y series temporales para flujos vehiculares. Esta aproximación metodológica integral asegura que la solución desarrollada sea tanto técnicamente robusta como socialmente responsable.

El ciclo culmina con la etapa **VISUALIZE/SHARE (Comunicar)**, donde se transforman los hallazgos complejos en representaciones accesibles. Se desarrollan visualizaciones estáticas creadas en Python (Matplotlib/Seaborn) para comunicar resultados específicos. Este componente es crucial para garantizar que los insights derivados del análisis sean comprendidos y aplicados efectivamente en la implementación del sistema de semáforos inteligentes.

Timeline (planificación)

A continuación se muestra la planificación desarrollada incluyendo su respectivo diagrama de Gantt.

Fase 1: Planificación y Revisión (Febrero - Marzo)

- **Febrero 26 - Marzo 7:** Definir objetivos específicos y alcance del proyecto.
 - Delimitar el enfoque en alguna calle o av. como Gran Vía y la interacción con vehículos autónomos.
 - Identificar las variables a considerar (volumen de tráfico, tipos de vehículos, peatones, etc.).

- Formular preguntas de investigación.
- **Marzo 8 - Marzo 21:** Revisión exhaustiva de la literatura.
 - Buscar artículos, proyectos e informes relevantes sobre semáforos inteligentes, IA en la gestión del tráfico y big data en movilidad urbana.
 - Profundizar en los proyectos KI4LSA y HumanLight como referencia.
 - Contactar a instituciones como el Centro de Investigación del Transporte (TRANSyT) [Área de Investigación – Sistemas de movilidad sostenibles, inteligentes e integrados – Centro de Investigación del Transporte](#) o el Instituto Universitario de Investigación del Automóvil (INSIA) [Instituto Universitario de Investigación del Automóvil Francisco Aparicio Izquierdo](#) para obtener información y asesoría.
- **Marzo 22 - Marzo 31:** Selección de bases de datos y herramientas.
 - Identificar las bases de datos públicas con información relevante
 - Evaluar herramientas de simulación de tráfico como SUMO y plataformas de análisis de datos.
 - Evaluar la tecnología que vamos a usar FML.

Fase 2: Desarrollo y Simulación (Abril - Mayo)

- **Abril 1 - Abril 15:** Diseño del modelo de semáforo inteligente.
 - Definir la arquitectura del sistema y los algoritmos de IA a utilizar.
 - Considerar la integración con vehículos autónomos y la optimización multimodal.
- **Abril 16 - Mayo 2:** Implementación del modelo en la herramienta de simulación.
 - Configurar el entorno de simulación con las características de la Gran Vía.
 - Integrar los datos de tráfico y movilidad obtenidos de las bases de datos.
- **Mayo 3 - Mayo 16:** Ejecutar simulaciones y analizar resultados.

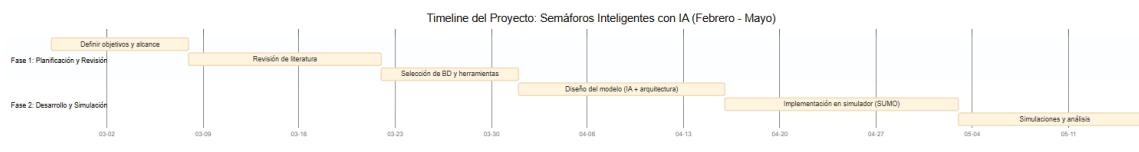


Figura 1: Diagrama de Gantt

2. Marco teórico

En este capítulo, se hace una introducción a las tecnologías de gestión de bases de datos, al machine learning y a la visión artificial, conocimientos esenciales para la comprensión del desarrollo del proyecto.

2.1. Tecnologías de Gestión de Bases de Datos

Es importante mencionar la división de tipos de bases de datos que existen ya que de aquí parte la decisión de depositar la información dependiendo del origen de la información.

Las bases de datos no relacionales (NoSQL) representan un paradigma alternativo al modelo relacional tradicional, diseñado específicamente para manejar grandes volúmenes de datos no estructurados o semiestructurados con esquemas flexibles. A diferencia de las bases SQL que usan tablas rígidas, los sistemas NoSQL emplean diversos modelos de datos como documentos (JSON/BSON en MongoDB), pares clave-valor (Redis), columnas anchas (Cassandra) o grafos (Neo4j), permitiendo escalabilidad horizontal mediante arquitecturas distribuidas. Estas bases sacrifican algunas propiedades ACID (como consistencia fuerte) en favor del teorema CAP, donde priorizan disponibilidad y tolerancia a particiones, implementando eventual consistency mediante mecanismos como quórum de escritura o replicación multi-nodo. Su arquitectura sin esquema fijo (schema-on-read) las hace ideales para big data, IoT o aplicaciones web escalables, donde el volumen, velocidad y variedad de datos superan las capacidades de las bases relacionales tradicionales.

Por otro lado, las bases de datos relacionales como SQL Server son sistemas fundamentados en el modelo relacional que organizan la información en tablas interconectadas mediante relaciones lógicas. Estas tablas almacenan datos en estructuras bidimensionales compuestas por filas (registros) y columnas (campos), donde cada tabla representa una entidad específica (como clientes o productos) y las relaciones entre ellas se establecen mediante claves primarias y foráneas. SQL Server implementa este modelo con un estricto esquema predefinido (schema-on-write) que exige declarar la estructura de datos antes de su uso, garantizando consistencia e integridad.

El núcleo de SQL Server se basa en el motor de base de datos (SQLOS), que incluye componentes críticos como el Query Optimizer para generar planes de ejecución eficientes, el Storage Engine que gestiona el almacenamiento físico en páginas de 8KB (archivos .mdf para datos y .ldf para logs), y el Buffer Pool que

cachea datos en memoria RAM para optimizar el rendimiento. La arquitectura sigue una jerarquía que va desde la instancia (que puede alojar múltiples bases de datos) hasta los filegroups y archivos físicos, permitiendo una administración granular de los recursos de almacenamiento.

SQL Server soporta el lenguaje SQL estándar con extensiones propietarias, dividiendo sus operaciones en tres categorías principales: DDL (para definir estructuras como tablas e índices), DML (para manipular datos con consultas SELECT, INSERT, etc.) y DCL (para gestionar permisos de acceso). Un aspecto clave es su cumplimiento de las propiedades ACID, donde destaca el uso de Write-Ahead Logging (WAL) para garantizar durabilidad, y mecanismos como los niveles de aislamiento (READ COMMITTED, SERIALIZABLE) para manejar transacciones concurrentes sin comprometer la consistencia.

Entre sus ventajas técnicas destacan su alto rendimiento para cargas OLTP, gracias a índices clustered y columnstore, y funciones avanzadas como AlwaysOn para alta disponibilidad. Sin embargo, presenta limitaciones en escalabilidad horizontal frente a alternativas NoSQL, y su modelo de licenciamiento por núcleo puede resultar costoso para implementaciones a gran escala. En proyectos como el sistema de semáforos inteligentes, SQL Server sería ideal para modelar datos estructurados de tráfico, ejecutar consultas complejas con joins entre tablas de sensores e intersecciones, e integrarse con herramientas de análisis como Power BI, aunque requeriría una planificación cuidadosa de recursos para manejar picos de carga eficientemente.

2.2. Machine Learning

En esta sección se va a hacer una breve introducción a los principales conceptos relacionados con la inteligencia artificial (IA) necesarios para comprender parte de la solución adquirida.

Dentro del campo de la inteligencia artificial se encuentra la rama denominada Machine Learning [9]. El aprendizaje automático (ML) consiste en la creación de

modelos matemáticos que permiten a las máquinas aprender de los datos sin necesidad de ser explícitamente programadas. Por ejemplo, una red neuronal es uno de los algoritmos más conocidos y utilizados en el ámbito del machine learning. Al procesar grandes volúmenes de datos, estas máquinas pueden identificar patrones y hacer predicciones.

La estadística juega un papel fundamental en el ML, ya que es esencial para entender los datos empleados en el entrenamiento de los modelos y para interpretar los resultados obtenidos.

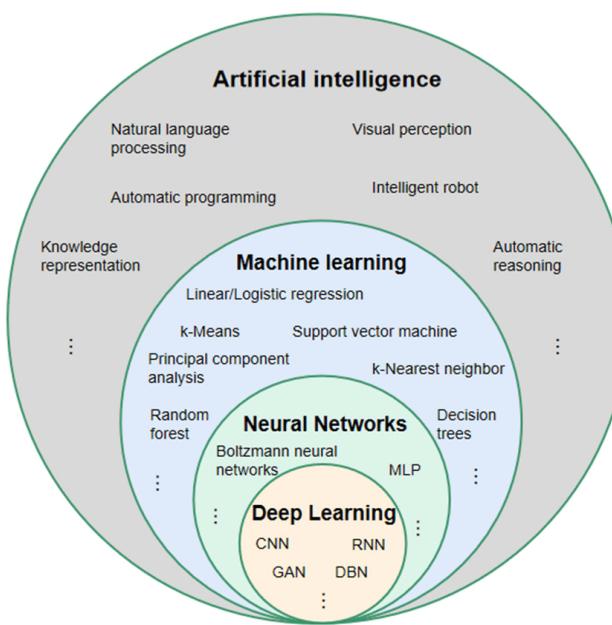


Figura 2: Esquema de lo que engloba Inteligencia Artificial

En este proyecto se ha empleado la técnica denominada Random Forest [10]. Un Random Forest en regresión es un modelo de aprendizaje automático que se utiliza para predecir valores numéricos continuos. Se basa en la construcción de múltiples árboles de decisión, cada uno entrenado con diferentes subconjuntos aleatorios de los datos y características. A diferencia de un solo árbol de decisión, que puede ser muy sensible a pequeñas variaciones en los datos, el Random Forest combina las predicciones de todos los árboles para obtener un resultado más robusto y preciso. En el caso de la regresión, la predicción final se obtiene promediando las salidas de todos los árboles del conjunto. Esta técnica ayuda a reducir el sobreajuste y

mejora la capacidad de generalización del modelo, lo que la hace especialmente útil en problemas complejos con muchos datos y relaciones no lineales.

2.3. Visión artificial

En esta sección se hace una breve descripción de lo que es un sistema de visión artificial y sus etapas.

Un sistema de visión artificial es la unión de hardware y software que posibilita la captura y el procesamiento de datos de imágenes. Es decir, es el sistema que permite que las máquinas puedan emular la vista humana. El programa es capaz de procesar, analizar y medir información para controlar los procesos y tomar las decisiones correspondientes. En un sistema, se pueden diferenciar dos fases principales: obtención de imágenes y el análisis de estas. A continuación se van a explicar en detalle.

A pesar de contar con dos fases principales, la de análisis se puede dividir en otras cuatro categorías más. De esta forma, un ciclo completo de un sistema de visión consta de los siguientes pasos:

- Adquisición de imágenes (etapa **sensorial**), consiste en conseguir una imagen. Consiste en emplear sensores y tener la capacidad para digitalizar la señal producida por el sensor (imagen).
- Transformaciones y filtrado de imágenes (etapa de **preprocesado**) para arreglar en la imagen las imperfecciones generadas o no suprimidas por el hardware con operaciones de mejora de la imagen. Otro de los objetivos es acondicionar la imagen en función del objetivo que haya. Las operaciones más importantes que se pueden aplicar a la imagen se puede dividir en tres apartados: operaciones básicas, transformaciones geométricas y algoritmos generadores de ruido y filtrado.
- Aislamiento de los objetos de la imagen (etapa de **segmentación**) Segmentar la imagen es dividirla en zonas individualizadas y disjuntas, En otras palabras, consiste en separar las partes que la constituyen o los objetos que la forman y el fondo. Es importante conocer perfectamente los objetos puesto que la extracción de características dependerá en gran medida de la correcta detección de estos.

- Cálculo de características (etapa de **parametrización**) se hace una selección de rasgos. Esto significa que en función de lo segmentado anteriormente se van a extraer unas características concretas que definen el objeto: forma, textura, color, orientación, etc.
- Reconocimiento e interpretación (etapa de **clasificación**) En base a los datos extraídos anteriormente se debe asignar una etiqueta y en función al resultado, interpretar y tomar una decisión o actuar en consecuencia.

3. Cuerpo central del Trabajo de campo/Desarrollo

En esta sección se describe el desarrollo principal del proyecto siendo este el nexo de unión entre los objetivos y el análisis de resultados del sistema.

En el contexto de la movilidad urbana inteligente, este trabajo se centra en optimizar los semáforos de la calle Gran Vía de Madrid - la zona más crítica de la ciudad por su alta densidad peatonal y vehicular - utilizando inteligencia artificial (IA) para mejorar tanto el flujo de peatones como la seguridad vial, priorizando a los transeúntes sin comprometer la eficiencia del tráfico. A través de un algoritmo basado en datos en tiempo real como densidad peatonal, volumen vehicular, eventos especiales y condiciones del entorno, se propone un sistema de semáforos autónomos que se adapte dinámicamente a las necesidades cambiantes de este emblemático corredor urbano.

La hipótesis plantea que, mediante un entrenamiento supervisado y la automatización integral comenzando por Gran Vía - para luego replicar el modelo en otras zonas - se puede reducir la congestión, minimizar tiempos de espera y crear un espacio más seguro e inclusivo. Bajo esta hipótesis, la imagen siguiente,

muestra la distribución de ubicación de cámaras en todo Madrid (encontradas en la página del Ayuntamiento de Madrid), de las cuales, las que están marcadas con colores son las áreas de enfoque del desarrollo del trabajo.

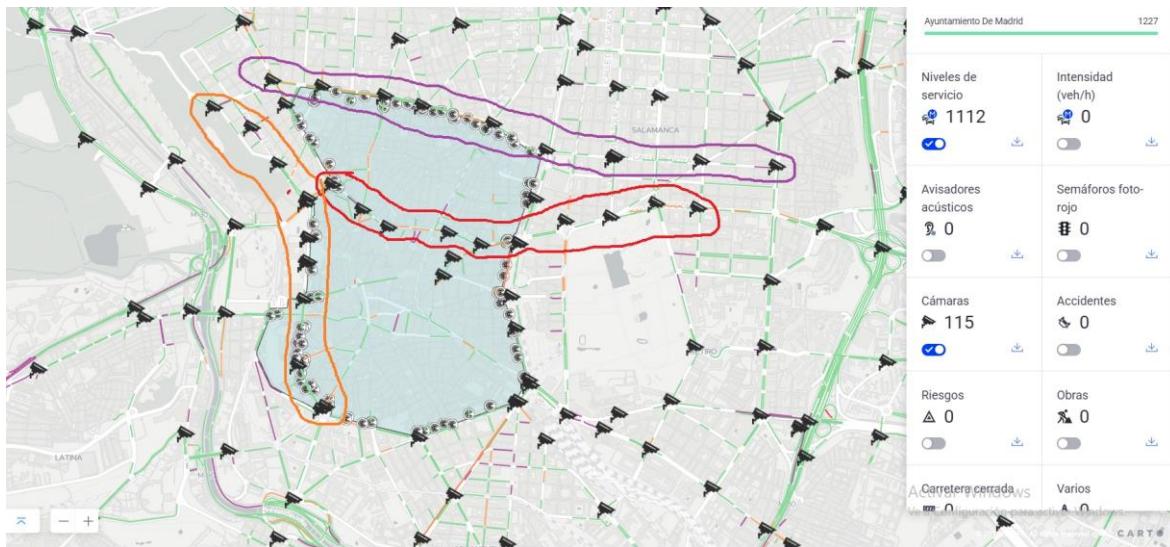


Figura 3: Distribución de cámaras de la ciudad de Madrid

Factibilidad del Proyecto: Desafíos Técnicos, Legales y Éticos

La implementación de un modelo de semáforos inteligentes autónomos en un entorno real presenta varios desafíos:

- **Técnicos:**

- La integración de diferentes tecnologías de sensores y sistemas de IA.
- La gestión de grandes volúmenes de datos en tiempo real.
- La garantía de la fiabilidad y la seguridad del sistema.

- **Legales:**

- La responsabilidad en caso de accidentes.
- La protección de la privacidad de los datos.
- El cumplimiento de las normativas de tráfico.

- **Éticos:**

- La equidad en la asignación de tiempos de verde.
- La discriminación potencial contra ciertos usuarios de la vía.
- La transparencia en la toma de decisiones del sistema.
- Los marcos éticos, como el utilitarismo y la deontología, pueden ayudar a guiar la toma de decisiones en los sistemas de semáforos autónomos. El utilitarismo se centra en maximizar el bien general, mientras que la deontología se basa en un conjunto de reglas predefinidas.

Se comenzó a trabajar en la detección de algunas variables que mientras está en desarrollo algunas irán cambiando, Hasta ahora, se detallan las siguientes:

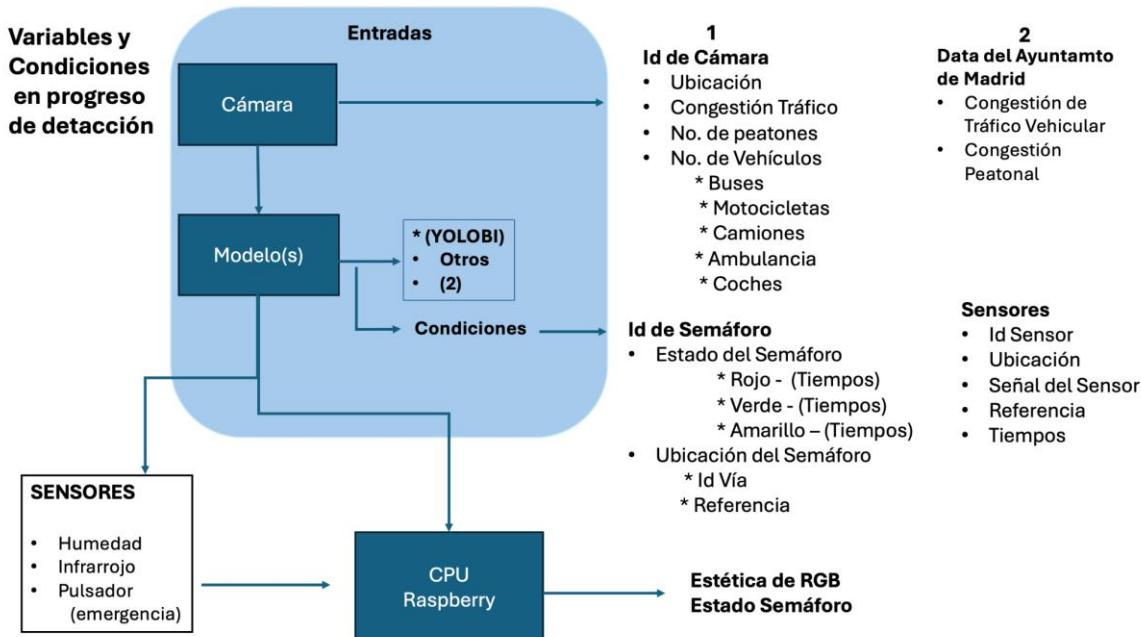
3.1. Descripción del Flujo del Sistema de Monitoreo de Tráfico

A continuación se describe lo que sería el flujo del sistema inteligente de gestión de tráfico urbano que integra múltiples componentes tecnológicos para el monitoreo y control del flujo vehicular y peatonal.

El sistema comienza con la captura de datos a través de cámaras de tráfico que alimentan uno o varios modelos de procesamiento, los cuales analizan las condiciones del tráfico en tiempo real y generan información sobre diversos parámetros como el estado de los semáforos, la congestión vehicular y peatonal, además de detectar diferentes tipos de vehículos incluyendo buses, motocicletas, camiones, ambulancias y coches.

La información procesada se estructura en diferentes categorías de datos que incluyen identificadores de cámara con detalles sobre ubicación, congestión de tráfico y conteo de peatones y vehículos por tipo, así como datos del Ayuntamiento de Madrid relacionados con la congestión vehicular y peatonal. De esta manera, el sistema será capaz de determinar los tiempos de su estado actual (rojo, verde, amarillo), junto con datos de ubicación y referencia de cada semáforo en la red vial.

Adicionalmente, el sistema integra una red de sensores especializados que incluyen sensores de humedad, infrarrojos y pulsadores de emergencia, todos conectados a una CPU Raspberry Pi que procesa esta información adicional y genera estadísticas sobre el estado de la red de semáforos mediante análisis RGB. Todo este ecosistema tecnológico permite una gestión integral del tráfico urbano, combinando detección visual inteligente, monitoreo ambiental y control automatizado de la infraestructura vial para optimizar el flujo de tráfico y mejorar la seguridad ciudadana. En la figura de abajo se muestra el flujo.



Esquema 1: Flujo del sistema inteligente

3.2. Software basado en bases de datos del Ayuntamiento de Madrid

Fuente de Datos

Recurso disponible con Información Real: Portal de datos abiertos del Ayuntamiento de Madrid

	Nombre del Datasource	Descripción del Datasource	Link del Web	Formato del Recurso
1	Aforo de Peatones y Bicicletas (Aforos de tráfico en la ciudad de Madrid permanentes)	Información relativa a la cantidad de peatones y bicicletas correspondientes a diferentes calles en diferentes días y horas de la semana.	Anexo 1: Estructura de Datos (Peatones) Anexo 2: Estructura de Datos(Bicicletas) [11]	XLS o CSV Para subirlo a la BD es requerido el CSV. Frecuencia de actualización: Anual
2	Aforos de tráfico en la ciudad de Madrid permanentes (Histórico)	Este conjunto de datos ofrece los datos de aforos de tráfico registrado en las 60 estaciones permanentes disponibles en la ciudad de Madrid.	Anexo 3: Estructura de Datos (Nombre del Datasource)) [12]	XLS o CSV Para subirlo a la BD es requerido el CSV. Frecuencia de actualización:: Mensual

3	Tráfico. Datos del tráfico en tiempo real (Mes vigente)	Este conjunto de datos ofrece los datos de aforos de tráfico registrado en las 60 estaciones permanentes disponibles en la ciudad de Madrid.	Anexo 4: Estructura de Datos (Nombre del Datasource) [13]	XML Proceso generado en programa Python para subir la data Actualización: 5 mins
4	Accidentes de Tráfico de la cd de Madrid	Accidentes de tráfico en la ciudad de Madrid registrados por la Policía Municipal.	Anexo 5: Estructura_ConjuntoDatos_Accidentes Visualización: [14], [15]	XLS o CSV Para subirlo a la BD es requerido el CSV. Frecuencia del Actualización: Anual
5	Tráfico. Ubicación de los puntos de medida del tráfico	Sirve para identificar los objetos georreferenciados que se suministran mediante ficheros Esri (SHP), en los conjuntos de datos de tráfico del Portal de Datos Abiertos del Ayuntamiento de Madrid	Anexo 6 Anexo 6.1: Manual básico para la visualización de la ubicación de dispositivos Anexo 6.2 Visualización desde Google Earth	.ZIP -> .SHP, .SHX, .DBF, .PRJ, Quantum Gis (QGIS).
6	Valores Climatológicos	Agencia Estatal de Meteorología	Valores Climatológicos -> Climatologías Diarias: Estación: Madrid Retiro [16] Anexo 7: Servicio del Banco Nacional de Datos Climatológicos	JSON Proceso Python para consumir la API Actualizacion: Diaria

Selección de Base de Datos Seleccionada y sus razones

3.2.1. Arquitectura de SQL Azure

Azure SQL Database [11] ofrece numerosas ventajas para proyectos de Big Data como el sistema de semáforos inteligentes de Madrid, destacando su perfecta integración con el ecosistema Microsoft. Al utilizar **Power BI**, permite conexiones nativas para visualizar datos de tráfico y métricas éticas en dashboards, mientras que **Azure Machine Learning** facilita el entrenamiento de modelos predictivos (como la previsión de congestión) directamente sobre los datos almacenados. Además, para una segunda fase del proyecto, **Azure Data Factory** simplifica la orquestación de procesos **ETL** para extraer información de sensores y cámaras.

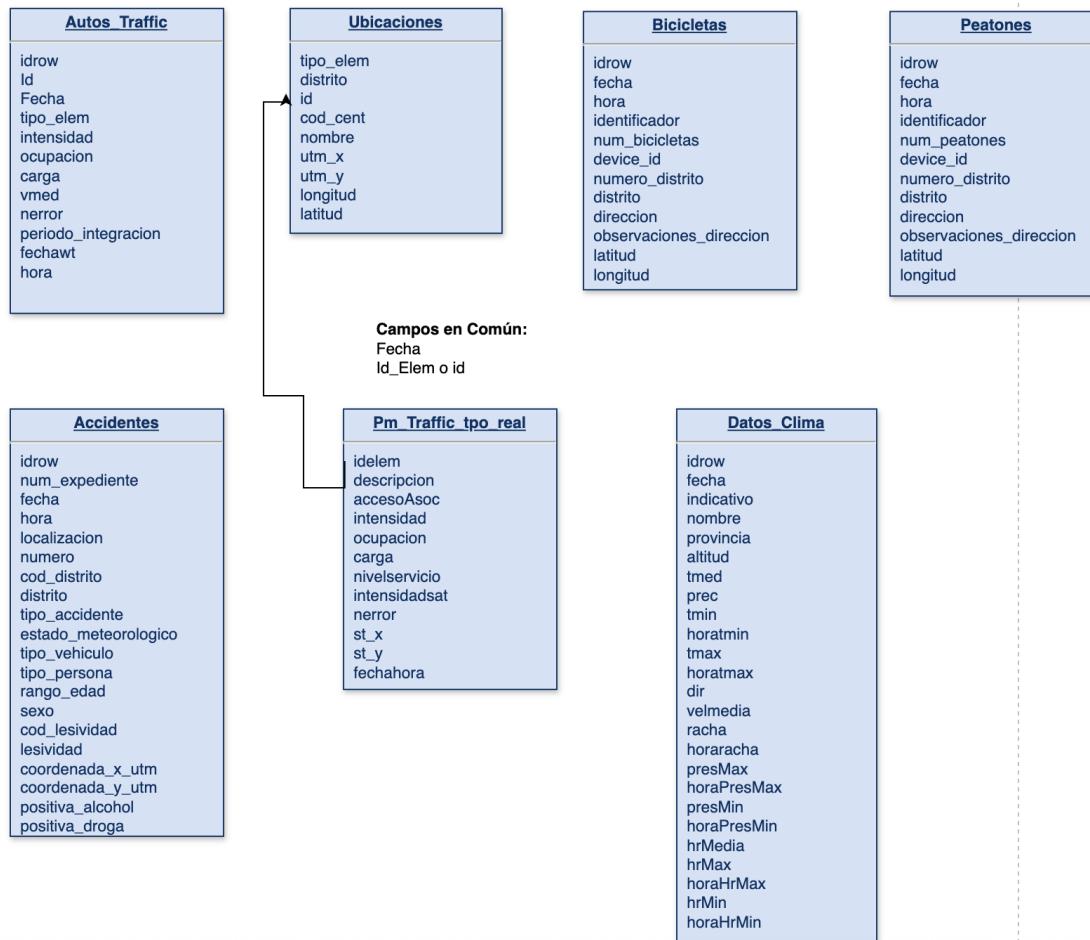
Su escalabilidad automatizada es ideal para ajustar recursos computacionales (DTUs) en horas pico, como durante eventos masivos en la ciudad.

En cuanto a seguridad, cumple con el **GDPR** mediante encriptación de datos en reposo y tránsito, crucial cuando se maneja información sensible de ciudadanos, e incluye auditorías integradas para garantizar transparencia en el acceso a los datos. También soporta formatos semiestructurados como **JSON/XML**, útil para datos de sensores o **APIs**, e incluso funciones de Graph DB para modelar relaciones complejas entre peatones, semáforos y calles.

Reduce la carga de mantenimiento gracias a parches automáticos gestionados por Microsoft y backups integrados que aseguran la recuperación ante fallos, como interrupciones en el sistema de semáforos.

En el esquema siguiente se muestra la base de datos diseñada para el cumplimiento de nuestros objetivos y procesamiento de predicciones detalladas en el análisis:

Base de Datos: Traffic_Congestion



Esquema 2: Diagrama de la Base de Datos

Una vez definida, se comenzó a popular cada tabla con su correspondiente información proveniente de los CSV bajados del Web del Ayuntamiento. Debido a la seguridad de la base de Datos y a la longitud del CSV ya que contiene todos los dispositivos de Madrid, se hizo una limpieza y filtrado a éstos archivos(en las siguientes páginas se detallan). Al analizar profundamente el contenido Histórico de tráfico permanente se descubrió que la zona más concurrente y más transitada por el centro de la ciudad, es Gran Vía, por el cual a partir de aquí se decidió enfocar el filtrado a los Elementos Ids de la zona de Gran Vía (Anexo 8)

3.2.2. Análisis de Tráfico Vehicular

Debido a lo explicado anteriormente, filtrando la data para crear un semáforo inteligente en **Gran Vía- Calle Clavel**, se optó por usar los datos del tráfico de Gran Vía como macro entorno y Clavel como micro entorno. En ese sentido, esta tabla de “**Autos_traffic**” contiene únicamente datos de Gran Vía y no de todo Madrid.

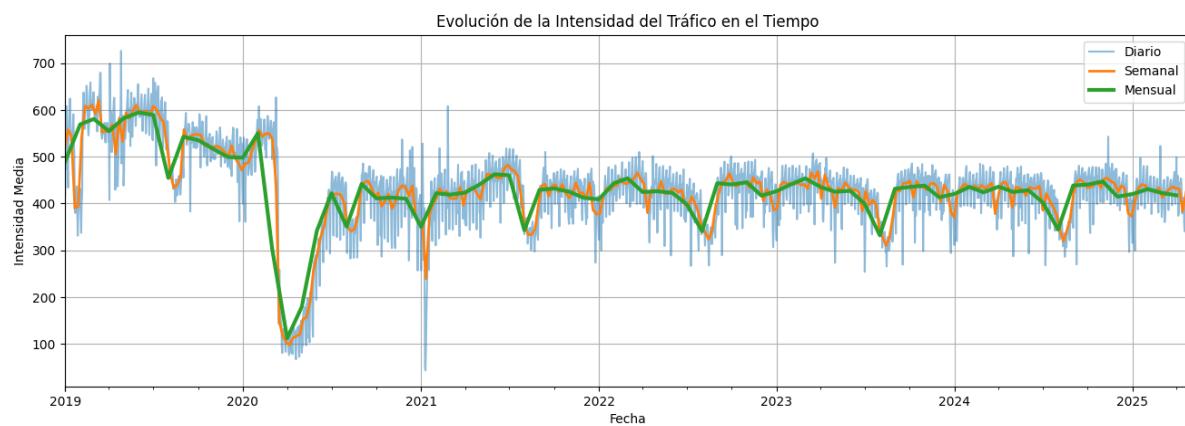


Figura 4. Evolución de la intensidad del Tráfico desde el 2019 en Gran Vía

Este gráfico muestra la evolución de la intensidad del tráfico a lo largo del tiempo desde 2019 hasta 2025, presentando tres series de datos: diaria (línea azul clara), semanal (línea naranja) y mensual (línea verde). Durante 2019, se observa una intensidad de tráfico relativamente alta y estable, con valores que oscilan entre 550 y 700 unidades, mostrando cierta variabilidad natural. El cambio más dramático ocurre a principios de 2020, cuando todas las series experimentan una caída abrupta y pronunciada, alcanzando mínimos históricos alrededor de 100-150 unidades de intensidad, lo que coincide temporalmente con el inicio de la pandemia de COVID-19 y las medidas de confinamiento implementadas globalmente.

Después de este mínimo histórico en 2020, el tráfico muestra una recuperación gradual pero estabilizándose en un nuevo nivel que se mantiene relativamente constante. Este nuevo nivel de equilibrio se sitúa aproximadamente entre 400 y 500 unidades de intensidad media, permaneciendo por debajo de los niveles pre-pandémicos de 2019. Las tres series de datos muestran patrones similares, con las medias semanales y mensuales proporcionando una visión más suavizada de las tendencias que los datos diarios, que presentan mayor volatilidad y fluctuaciones naturales día a día.

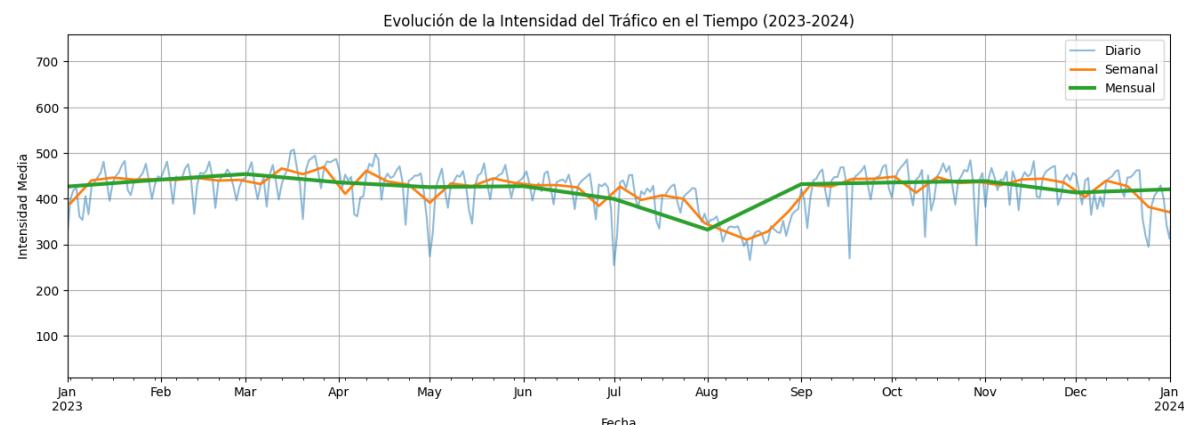


Figura 5: Intensidad de Tráfico en los años 2023-2024

Este gráfico demuestra la regularidad diaria explicada en el gráfico anterior sólo habiendo una excepción en Agosto del 2023 que supone obras de mantenimiento o de peatonalización en esas fechas.

Mapa de calor en Calle Clavel

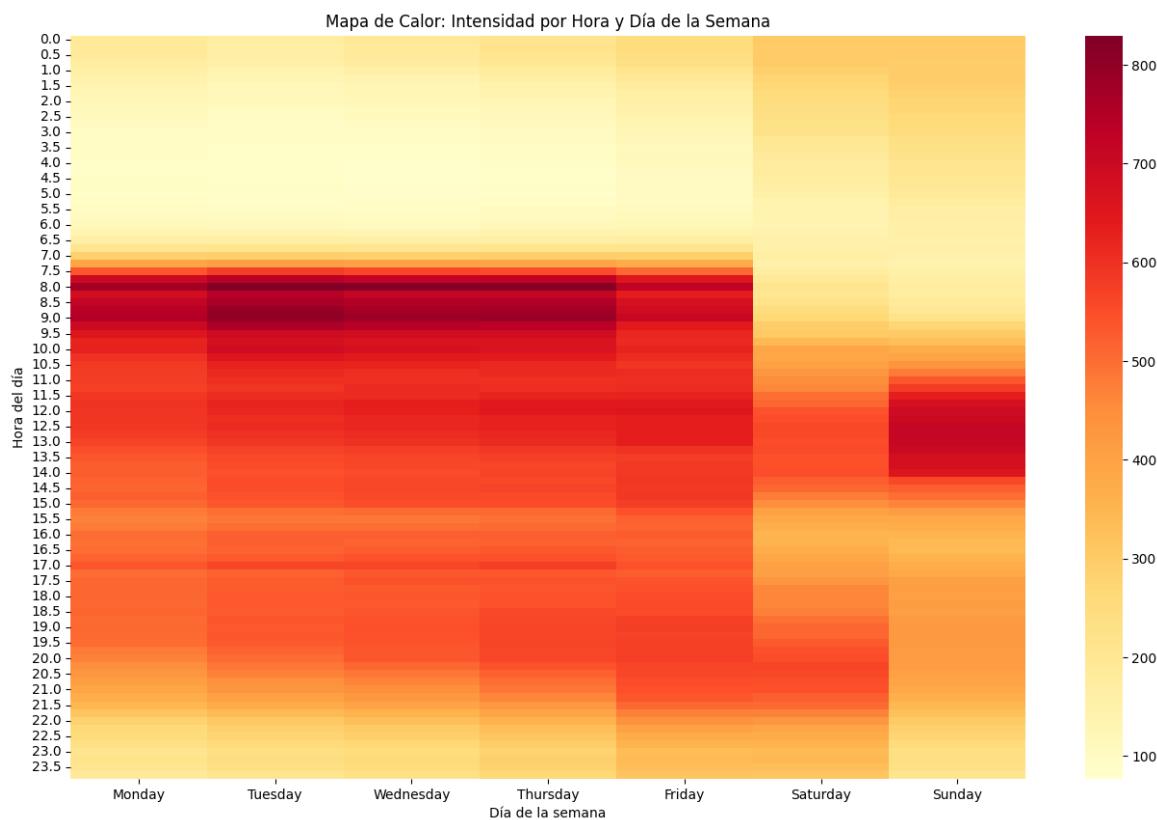


Figura 6. Mapa de calor en calle Clavel

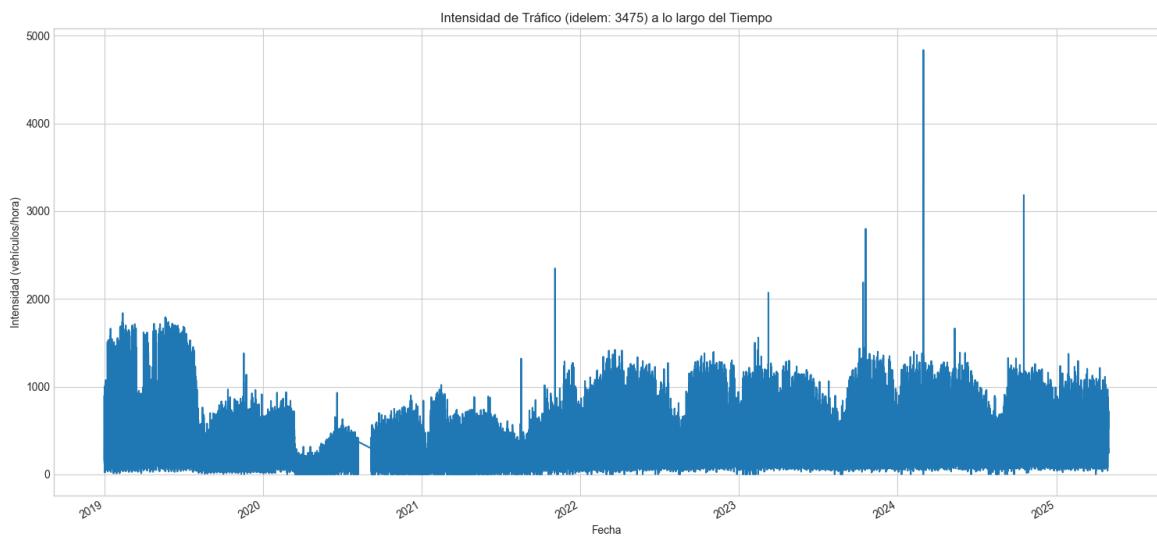


Figura 7. Intensidad por año en calle Clavel

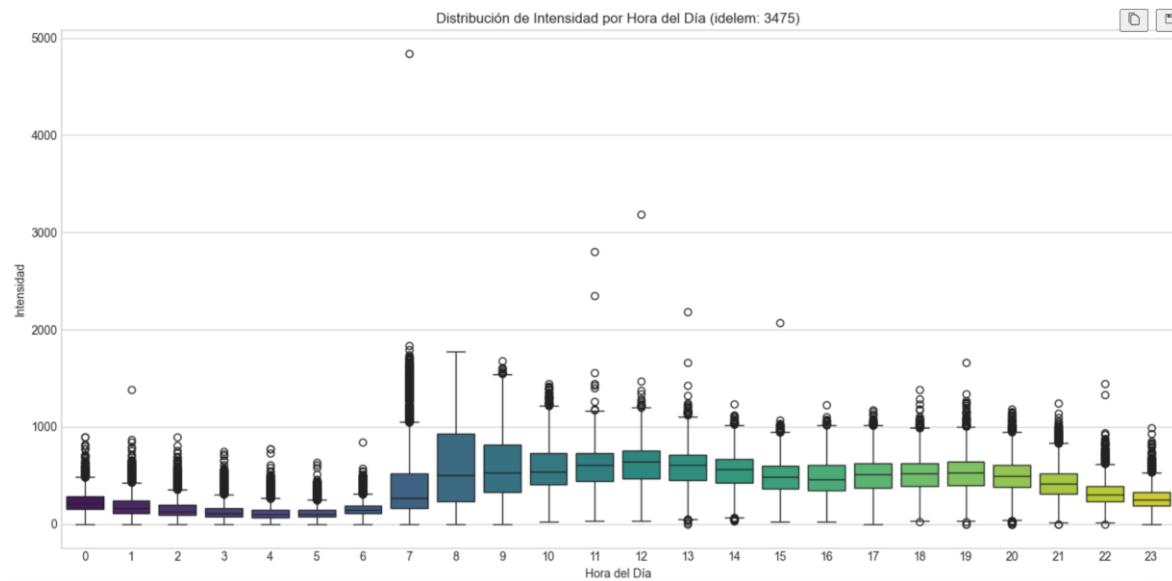


Figura 8. Intensidad Por hora, calle Clavel

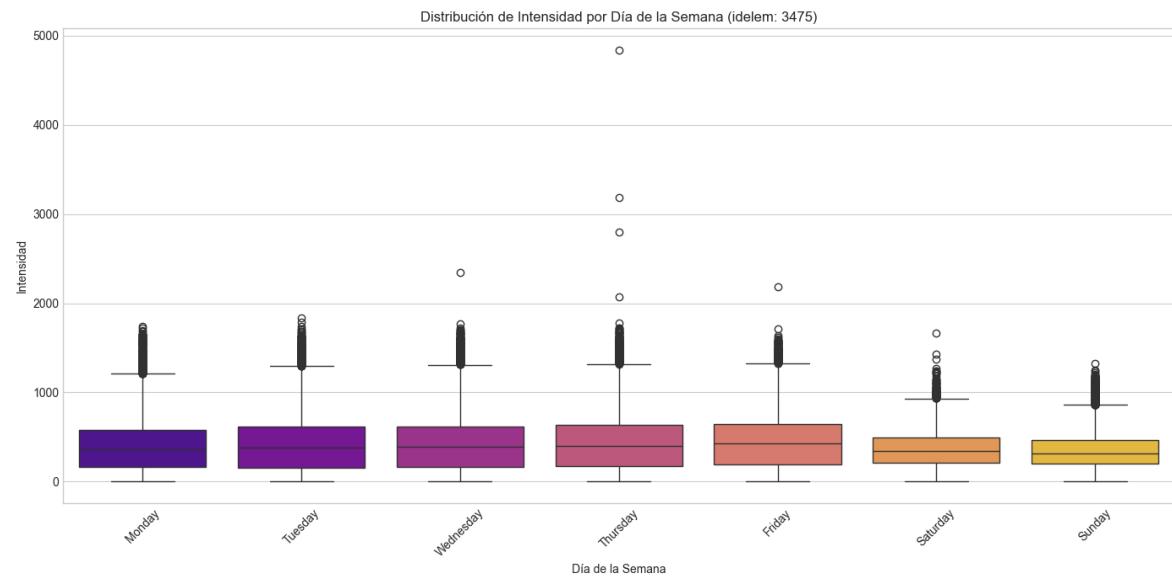


Figura 9. Intensidad Por día de la semana, calle Clavel

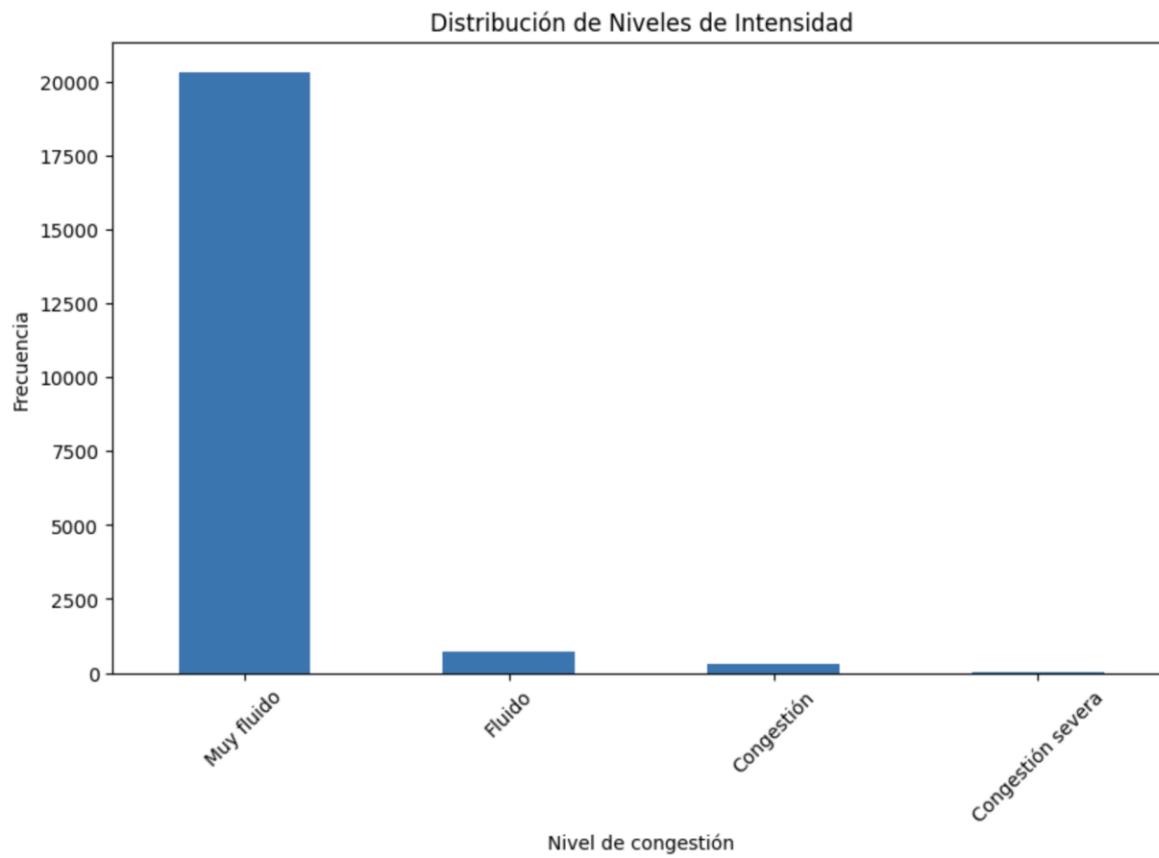


Figura 10. Niveles de intensidad

Densidad de Tráfico en Tiempo Real en Madrid, Gran Vía y Clavel (en ambos sentidos de dirección)

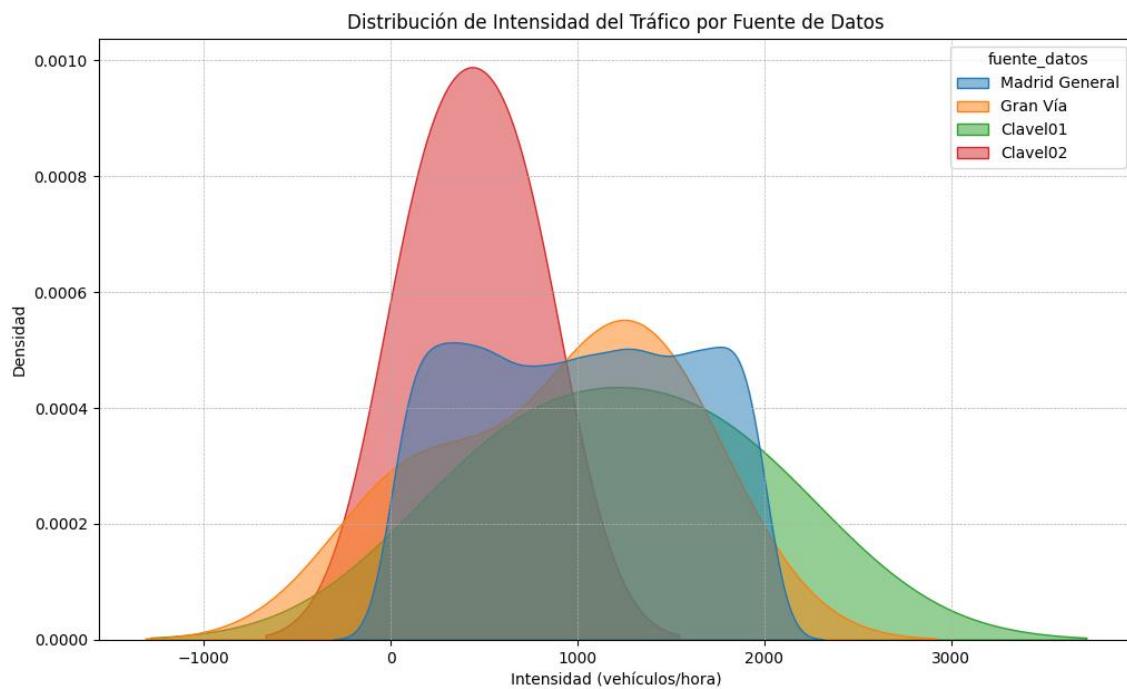


Figura 11. Distribución de Intensidad del Tráfico

Intensidad de Tráfico en Tiempo Real en Madrid, Gran Vía y Clavel (en ambos sentidos de dirección)

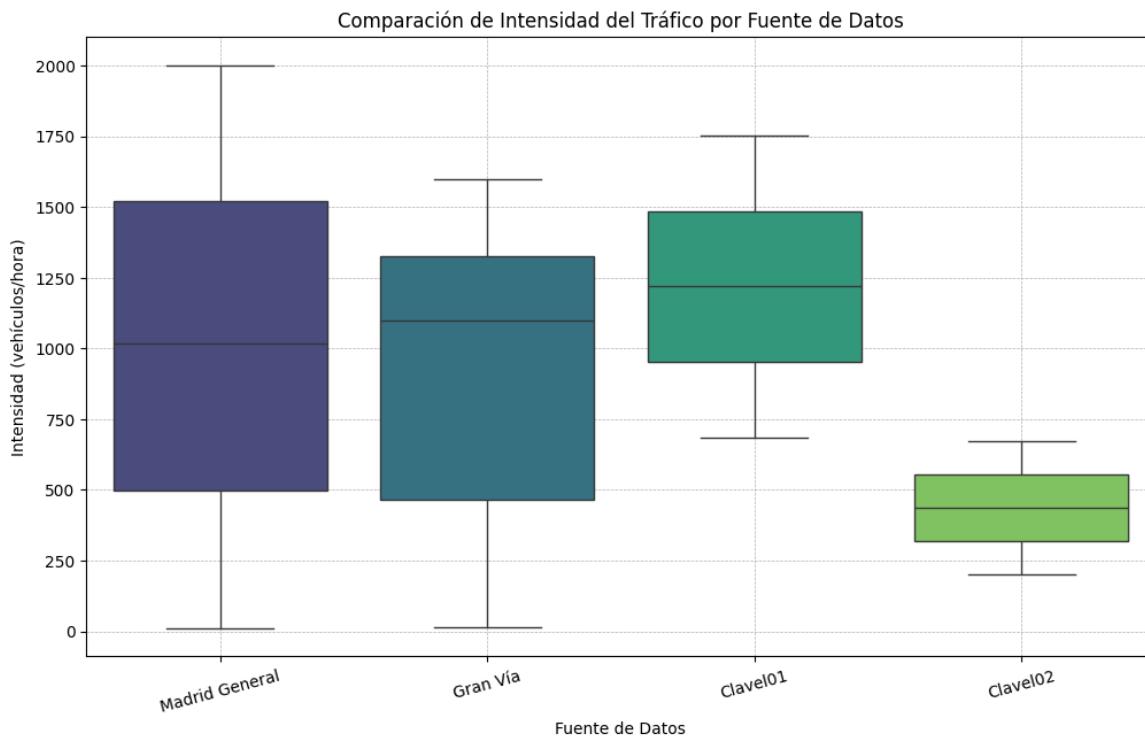


Figura 12. Comparativo intensidad de Tráfico por Localización

3.2.3. BBDD tráfico peatonal

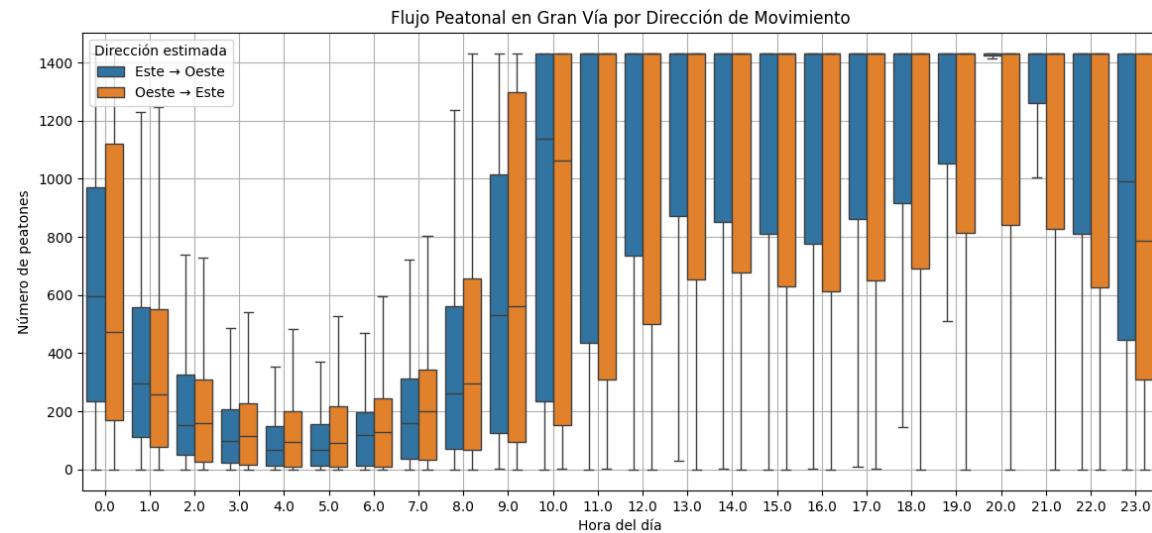


Figura 13. Flujo Peatonal en Gran Vía por dirección de movimiento

PROCESAMIENTO DE NA'S (Nulos)

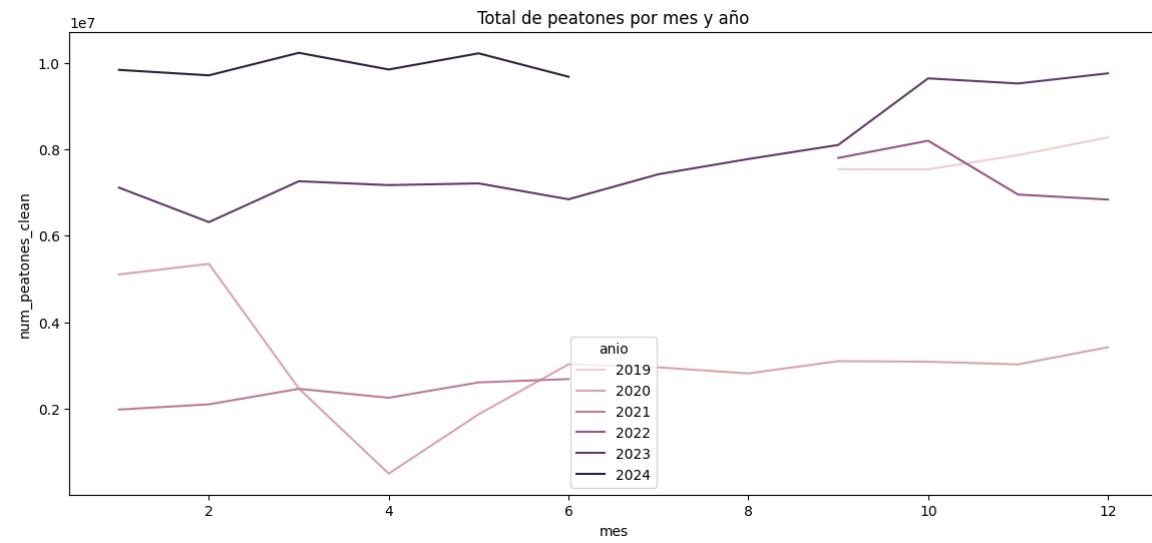


Figura 14. Total de Peatones por mes y año

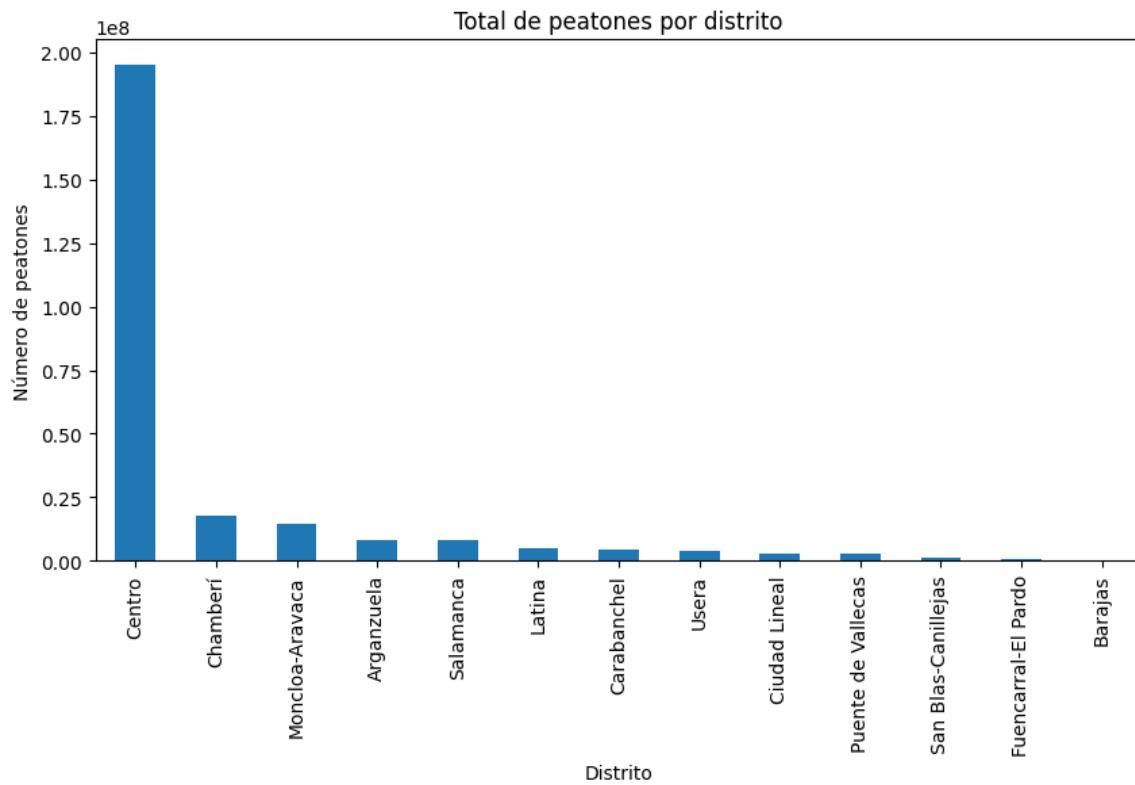


Figura 15. Total de Peatones por distrito

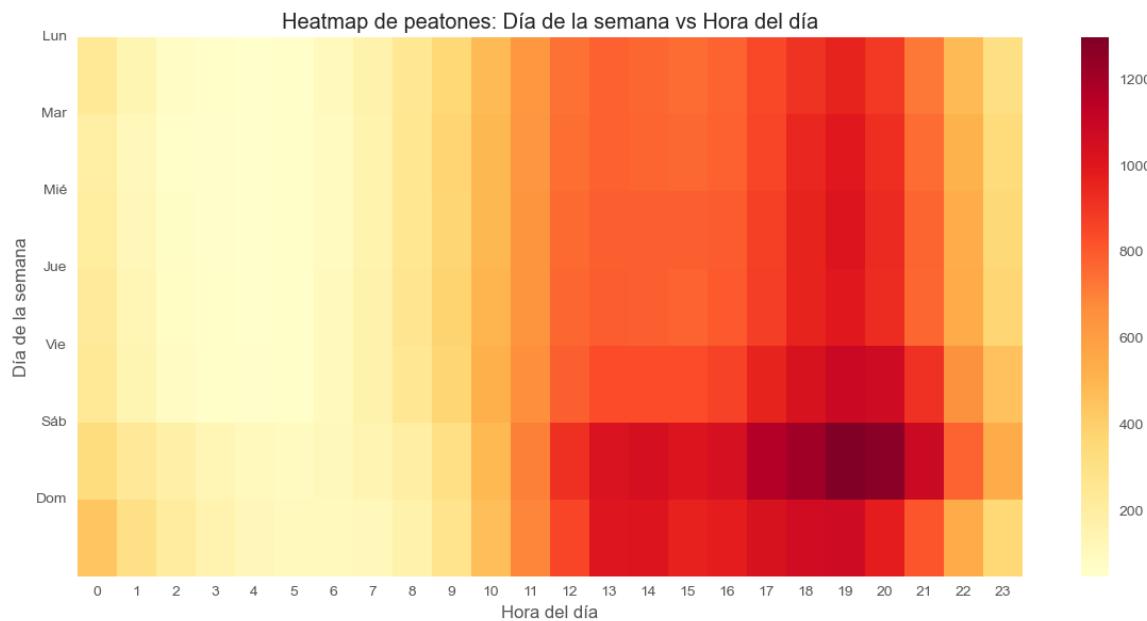


Figura 16. Heatmap de peatones: Día de la semana vs Hora del día

Este gráfico ilustra la densidad de tránsito peatonal a lo largo de la semana y las horas del día. La intensidad del color refleja la concentración de peatones, como las horas pico de días laborables (por ejemplo, entre las 8-9 a.m. y las 5-7 p.m.) son de mayor aglomeración, mientras que las madrugadas (0-5 a.m.) muestran una actividad mínima. Los fines de semana (Sábado y Domingo) presentan un patrón distinto, con mayor movimiento durante las tardes o noches debido a actividades recreativas. Esta visualización resulta valiosa para identificar tendencias de movilidad urbana, optimizar recursos en transporte o comercios, y planificar infraestructuras peatonales según la demanda horaria y semanal.

Análisis de mayor flujo peatonal

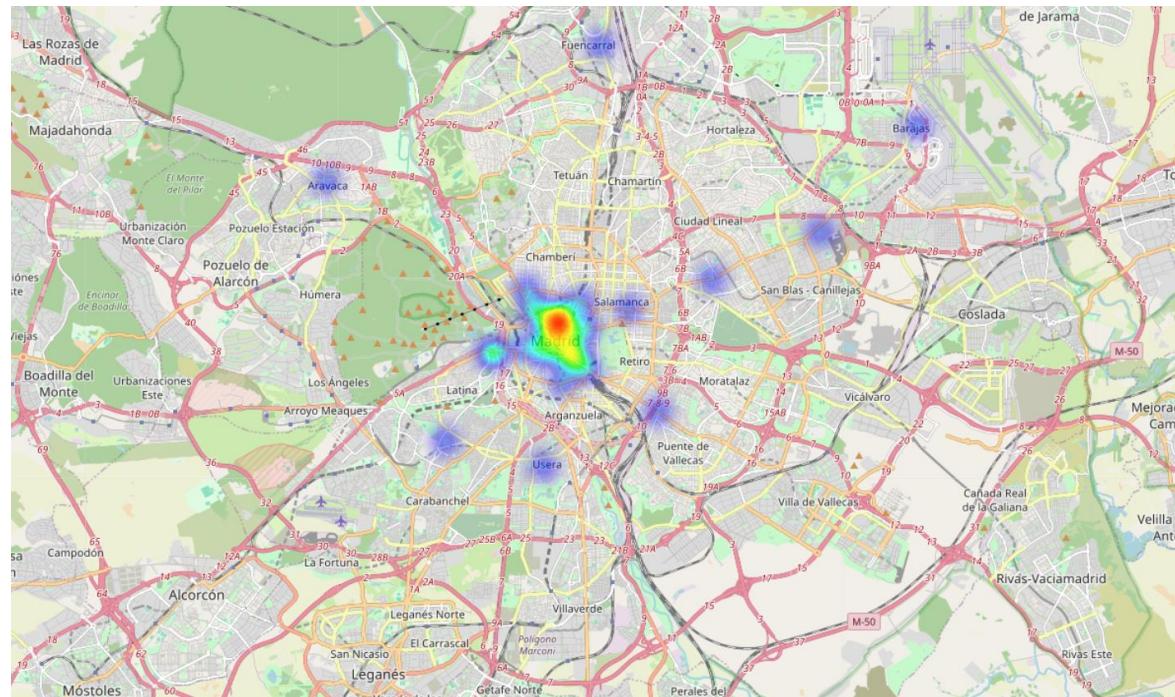


Figura 17. Mayor flujo peatonal en zona centro

3.2.4. BBDD bicicletas

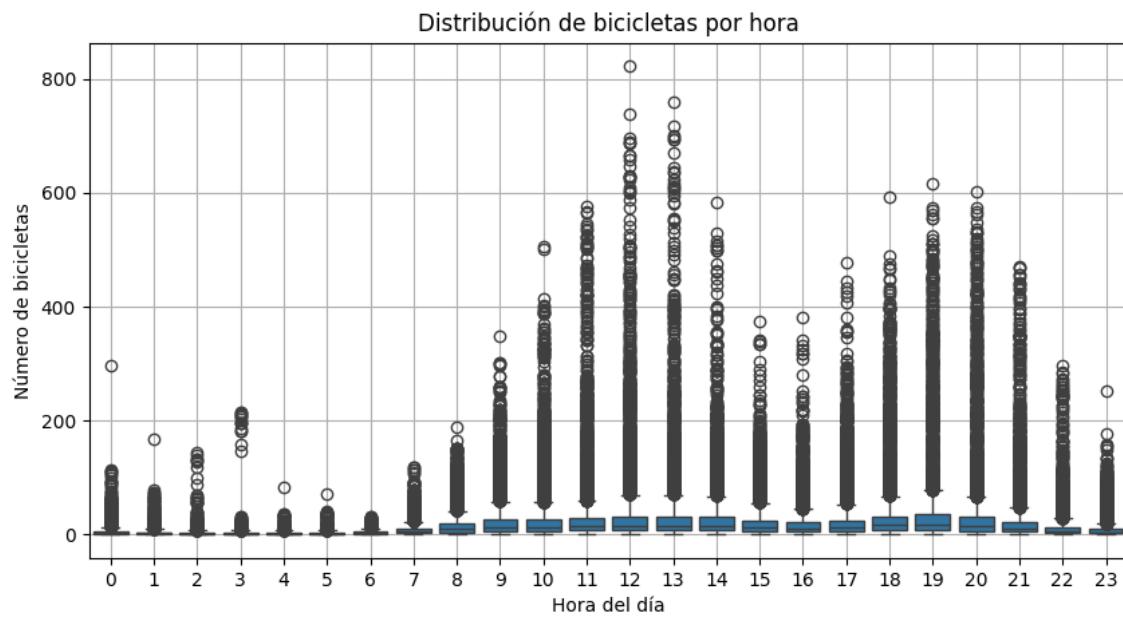


Figura 18. Distribución de bicicletas por hora del día

La imagen representa la cantidad de bicicletas utilizadas o disponibles en diferentes horas del día. El gráfico probablemente ilustra patrones de uso, como picos durante las horas de mayor actividad (como las horas pico de la mañana o la tarde) y

disminuciones en períodos de menor demanda (como la madrugada). Este tipo de visualización es útil para analizar tendencias y optimizar servicios relacionados con bicicletas, como sistemas de alquiler o infraestructura ciclista.

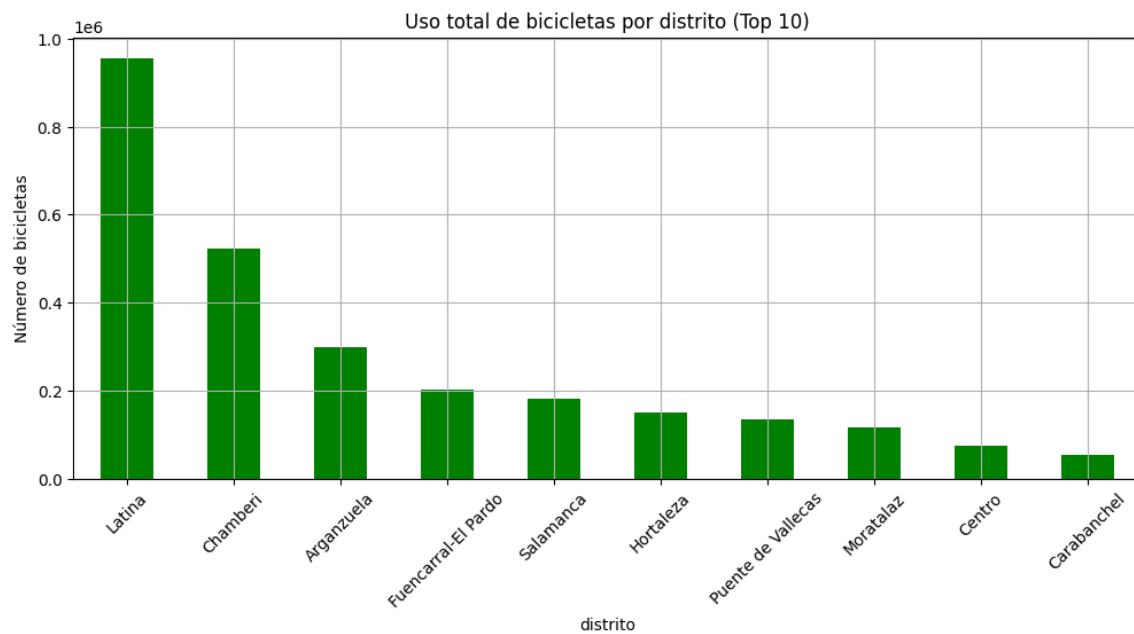


Figura 19. Bicicletas por Distrito

El distrito con mayor uso de Bicicletas es **Latina**, superando ampliamente las 950,000 bicicletas siguiéndole **Chamberí** por ser una de las zonas más céntricas.

3.2.5. BBDD accidentalidad

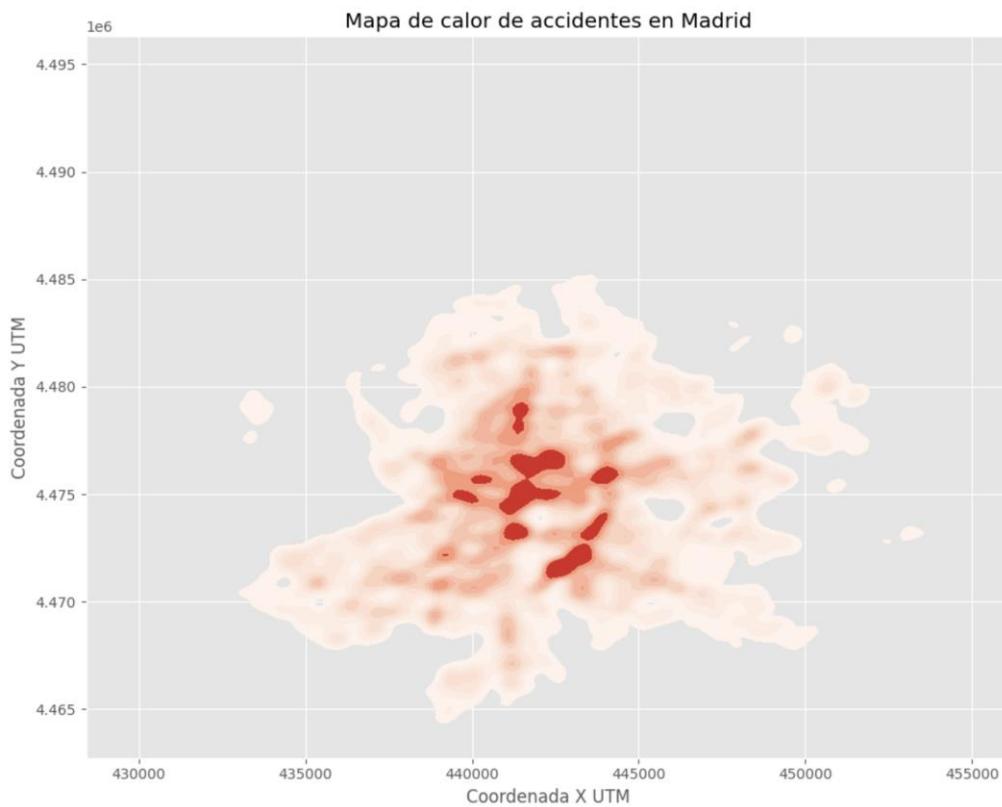


Figura 20. Mapa de calor de accidentes en Madrid

Como se ve en el mapa de calor ésta representa la distribución geográfica de los **accidentes** coincidiendo que, una de las áreas de mayor concentración de Accidentes (rojo intenso) es en la calle de **Gran Vía**.

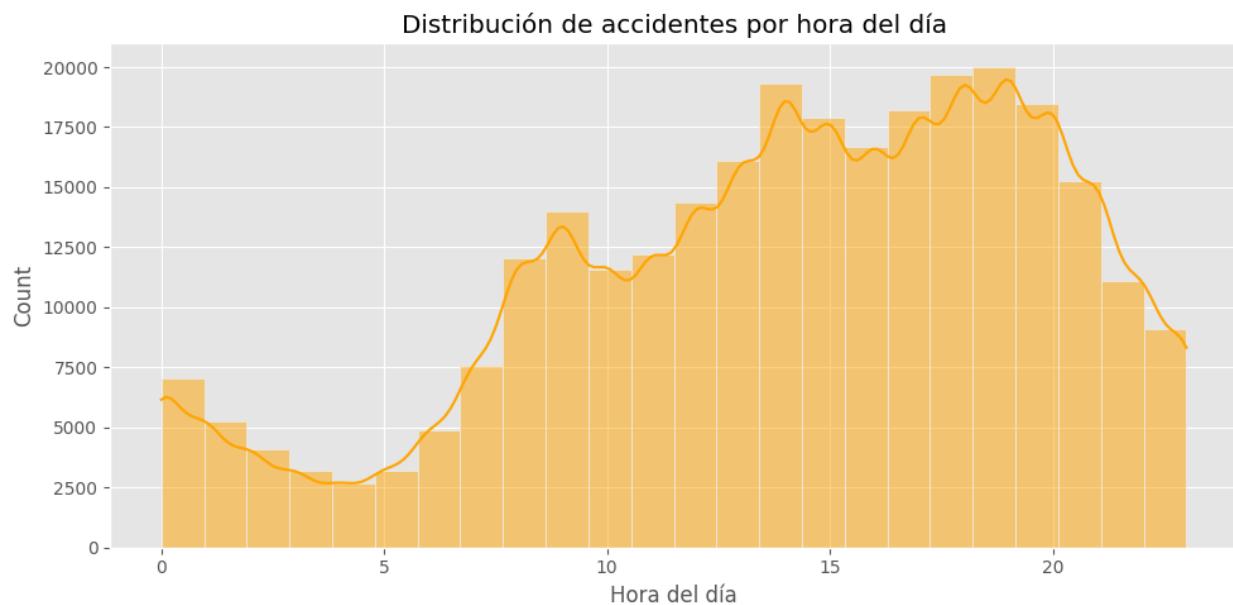


Figura 21: Accidentes por hora en Madrid

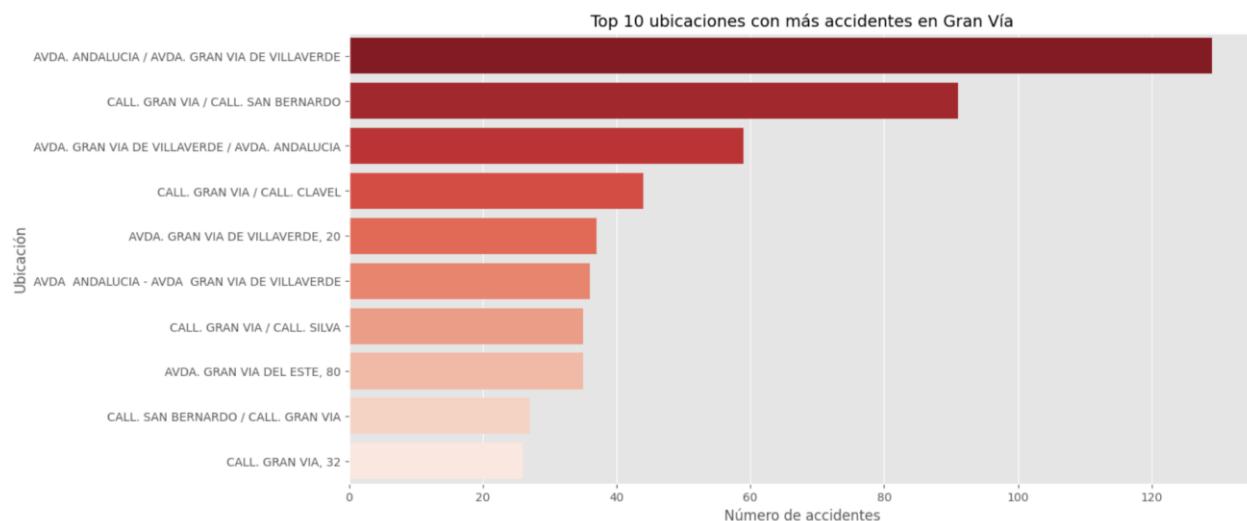


Figura 22: Accidentes por Ubicación en Gran Vía



Figura 23: Accidentes por Año en Gran Vía

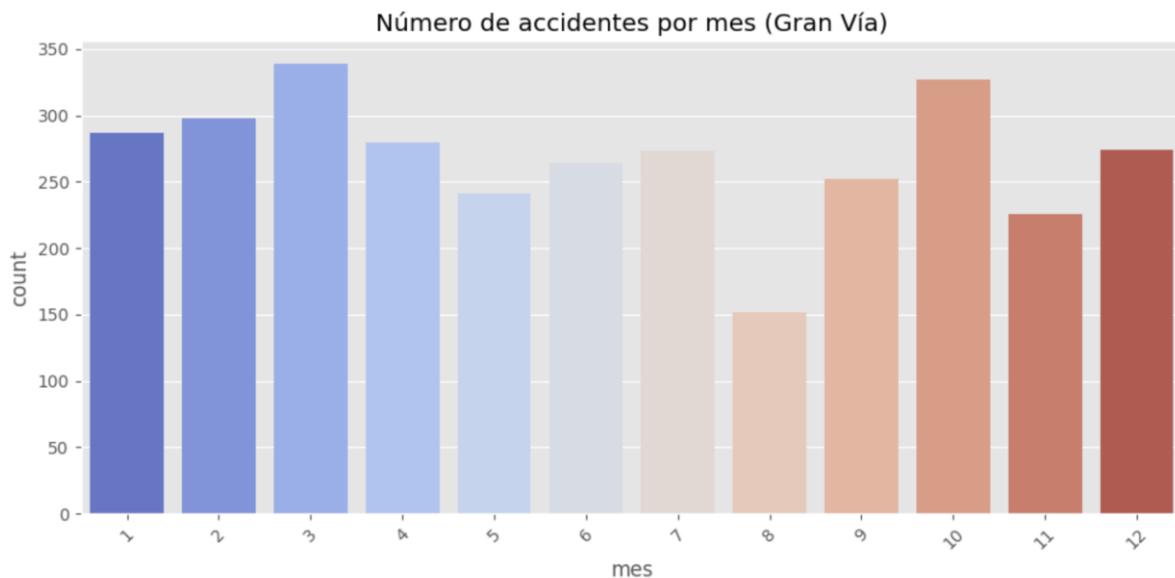


Figura 24. Accidentes por Hora Gran Vía

Densidad de accidentes por hora en Clavel, Gran Vía

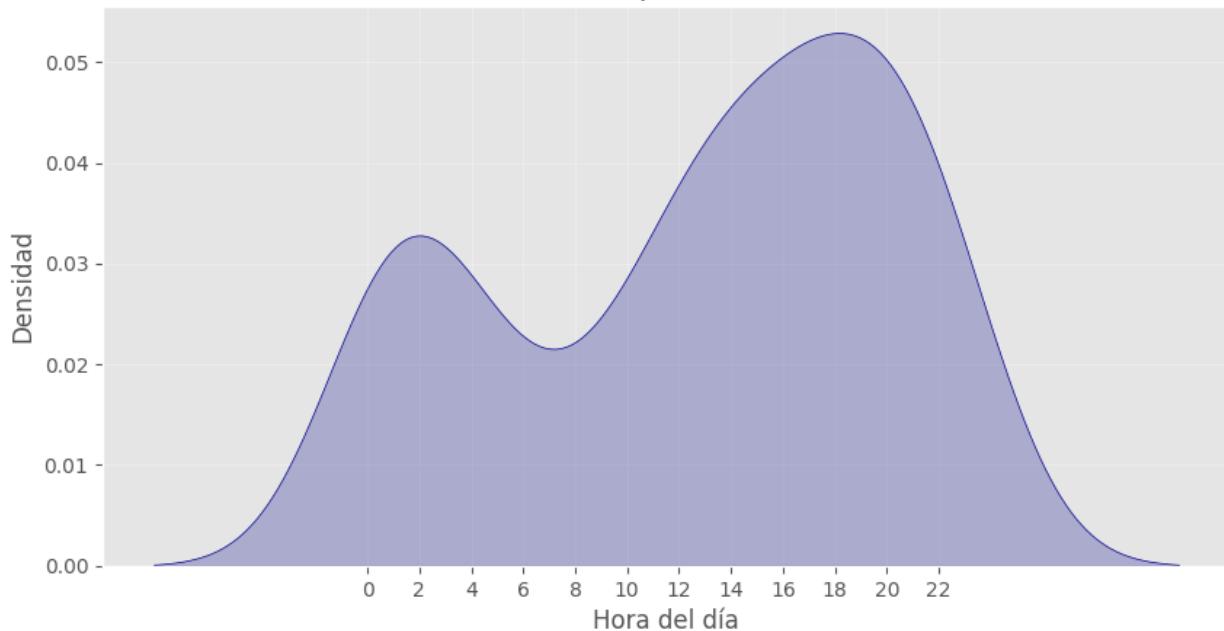


Figura 25. Densidad de accidentes por Hora en calle Clavel

Distribución de tipos de accidente en Clavel, Gran Vía

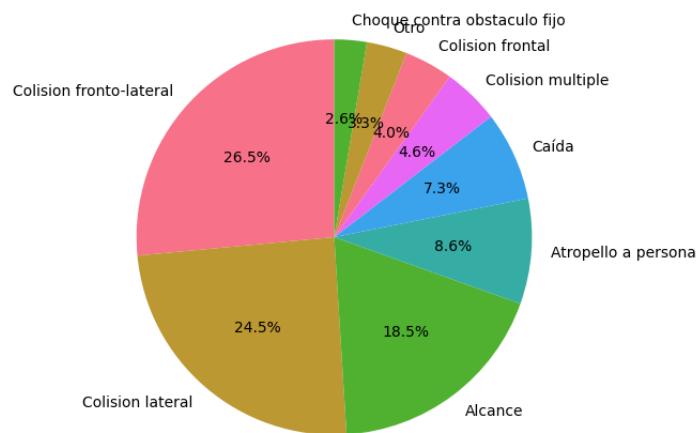


Figura 26. Tipos de Accidentes en calle Clavel

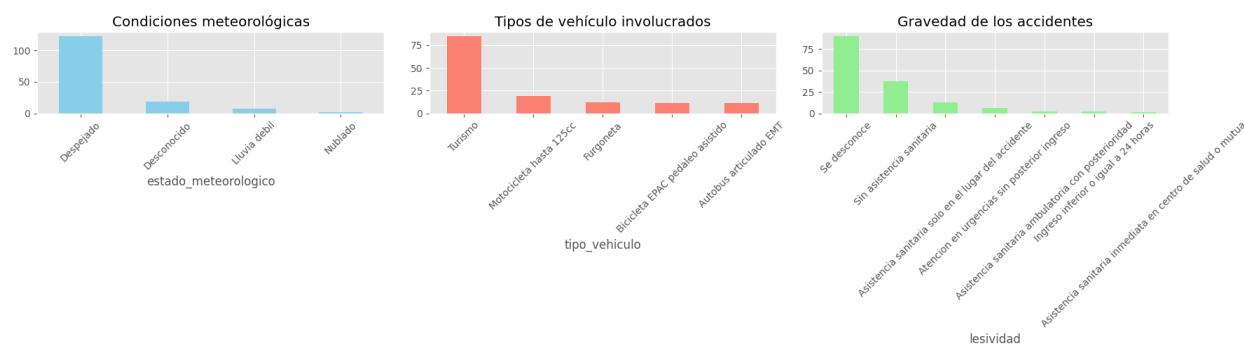


Figura 27. Diferentes variables en Accidentes de calle Clavel

Modelo predictivo

Tras analizar los datos y obteniendo las conclusiones descritas anteriormente

El modelo predictivo de tráfico propuesto utiliza el algoritmo Random Forest para abordar dos objetivos principales: la clasificación del nivel de servicio del tráfico y la predicción de valores numéricos como la intensidad vehicular, la ocupación y la carga. Este enfoque dual permite no solo entender el estado general del tráfico, sino también obtener estimaciones precisas que son fundamentales para la optimización de semáforos. El Random Forest es especialmente adecuado para este problema debido a su capacidad para manejar múltiples variables, su resistencia al sobreajuste y su habilidad para trabajar con datos incompletos o con ruido. Además, proporciona información valiosa sobre la importancia de cada variable, lo que ayuda a identificar los factores que más influyen en el flujo del tráfico.

Las variables de entrada incluyen datos temporales como la hora del día, el día de la semana y el mes, así como información específica del tráfico como la intensidad, la ocupación, la carga y la velocidad media. También se consideran variables contextuales, como el tipo de vía y el distrito, que pueden ser determinantes en el comportamiento del tráfico. Antes de entrenar el modelo, es crucial realizar un preprocesamiento de los datos, que implica la conversión de columnas como intensidad y ocupación a un formato numérico, manejar valores faltantes o erróneos, y crear características derivadas como promedios móviles o valores de

horas anteriores. Este paso asegura que los datos estén en un formato adecuado para el análisis.

Una vez preparados los datos, el modelo de clasificación predice el nivel de servicio del tráfico, categorizando en flujo, lento, con retenciones o congestionado. Paralelamente, el modelo de regresión estima valores numéricos exactos para la intensidad, la ocupación y la carga, lo que permite un ajuste más fino en la gestión de los semáforos. Estas predicciones se integran en un sistema de reglas que determina el tiempo óptimo de luz verde en los semáforos. Por ejemplo, en condiciones de tráfico fluido, se mantienen los tiempos estándar, mientras que en situaciones de congestión, se priorizan tiempos extendidos para aliviar la saturación.

La implementación del modelo en Python abarca desde la carga y limpieza de los datos hasta el entrenamiento y evaluación del modelo, utilizando herramientas como sklearn para el aprendizaje automático y matplotlib para la visualización de resultados. Para futuras mejoras, se puede incorporar datos externos como condiciones meteorológicas o eventos especiales, lo que podría aumentar la precisión del modelo. También se podría explorar el uso de modelos de series temporales avanzadas, como las LSTM, para capturar patrones temporales más complejos. Este enfoque integral no solo predice el estado del tráfico, sino que también sugiere acciones concretas para optimizar la movilidad urbana, adaptándose dinámicamente a las condiciones en tiempo real.

3.3. Software basado en visión computacional

En este apartado se describen los tres principales códigos empleados para el desarrollo del sistema: obtención de datos, análisis de datos y simulación de funcionamiento del sistema.

3.3.1. Obtención de imágenes reales (Web scraping)

En esta sección se describe el código que se ha empleado para la obtención de las imágenes del Ayuntamiento de Madrid. Se ha empleado una web del Ayuntamiento denominada Informo [18] y la técnica empleada es Web Scraping.

El código implementa tres funciones principales que trabajan de forma coordinada: `download_image()`, `zip_images()` y `main()`, cada una con un rol específico en el proceso de recolección de imágenes.

La función `download_image()` es el núcleo del sistema. Recibe como parámetro el ID de una cámara (como "01315" siendo esta la de especial interés) y construye la URL correspondiente añadiendo un parámetro `rand` basado en el timestamp actual para evitar caché. Utiliza la librería `requests` para descargar la imagen y la guarda localmente con un nombre estructurado como `camera_01315_20250507_1415.jpg`, donde se incluye el ID de cámara, fecha (año, mes, día) y hora (hora, minutos). La marca temporal se ajusta al uso horario de Madrid sumando una hora al UTC.

La función `zip_images()` se encarga de consolidar todas las imágenes descargadas. Recorre el directorio `camera_images` y comprime su contenido en un archivo ZIP llamado `camera_images.zip` usando la librería `zipfile`. Este paso es crucial para manejar eficientemente múltiples imágenes antes de su descarga masiva.

La función `main()` orquesta todo el proceso. Contiene la lista de cámaras (como "01315") y controla las iteraciones del sistema. En cada ciclo (5 por defecto), itera sobre todas las cámaras llamando a `download_image()` con una pausa de 2 segundos entre cada descarga para evitar saturación. Entre iteraciones completa espera exactamente 294 segundos (4 minutos y 54 segundos) para compensar el tiempo de ejecución y mantener intervalos precisos de 5 minutos.

La sección de configuración inicial define variables clave como `base_url_template` con el formato de URL base y `output_dir` para el almacenamiento local. El bloque final del código verifica si es ejecutado como script principal para iniciar el proceso llamando a `main()`, que a su vez invoca `zip_images()` al terminar todas las iteraciones, generando el archivo comprimido listo para descargar mediante la integración con Google Colab.

3.3.2. Detección de vehículos y coches en imágenes

En este apartado se describe el código que se emplea para la detección de elementos en las imágenes extraídas del Ayuntamiento de Madrid.

El código implementa un sistema de detección y conteo de vehículos y peatones utilizando el modelo YOLO, estructurado en una función principal **`detect_and_count()`** que realiza todo el procesamiento de imágenes.

La función `detect_and_count()` recibe como parámetro la ruta de una imagen y sigue un flujo bien definido. Primero verifica la existencia del archivo usando `os.path.exists()` y carga la imagen con `cv2.imread()`. Si la carga es exitosa, redimensiona la imagen a 1280x720 píxeles usando `cv2.resize()` para optimizar el procesamiento. Luego realiza la detección de objetos llamando al modelo YOLO con umbrales de confianza (0.4) e IoU (0.45) equilibrados.

El código inicializa contadores para vehículos (`vehicle_count`) y peatones (`pedestrian_count`), luego procesa los resultados de la detección. Para cada objeto detectado, obtiene su clase mediante `box.cls` y la traduce a texto usando `model.names`. Los vehículos se identifican cuando la etiqueta coincide con 'car', 'truck', 'bus' o 'motorcycle', mientras los peatones se detectan con la etiqueta 'person'. Cada detección válida incrementa los contadores correspondientes y se marca en la imagen con un rectángulo delimitador (bounding box) usando `cv2.rectangle()`, usando color verde para peatones y rojo para vehículos.

Finalmente, la función guarda la imagen anotada añadiendo '_detected' al nombre original mediante `cv2.imwrite()` y retorna los conteos junto con la ruta de la imagen procesada. La sección de ejemplo al final del código demuestra su uso, iterando sobre una lista de rutas de imágenes ('image_paths') y mostrando los resultados por consola, incluyendo los conteos y ubicación de las imágenes procesadas.

El código utiliza principalmente las bibliotecas 'ultralytics' para el modelo YOLO y 'OpenCV (cv2)' para el procesamiento de imágenes, mostrando un enfoque modular donde la función principal encapsula toda la lógica de detección y conteo, mientras la parte de demostración maneja la ejecución concreta y presentación de resultados.

3.3.3. Código de simulación

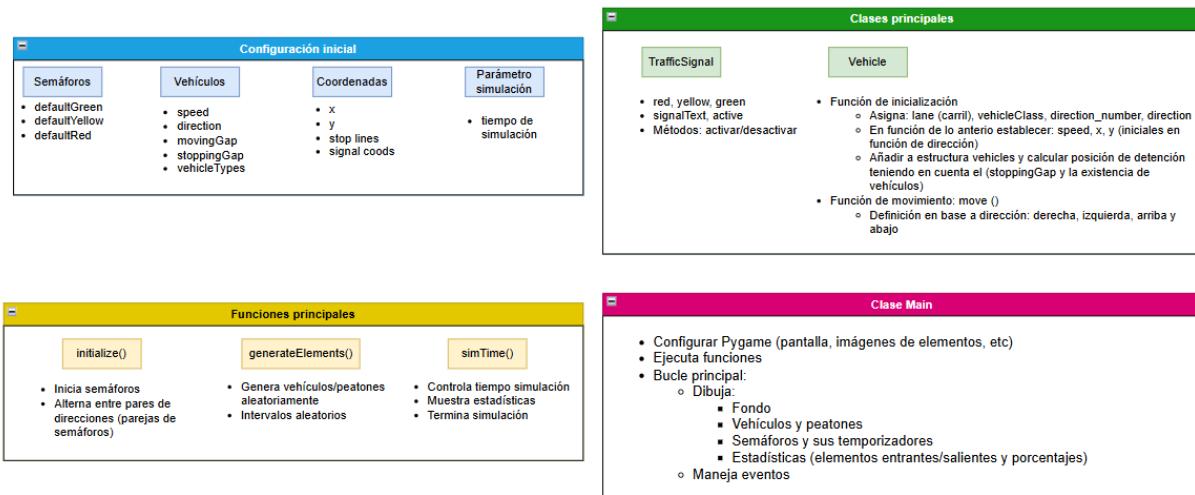
En esta sección se describe el código que se ha empleado adaptándolo al sistema de estudio para hacer una simulación del funcionamiento del sistema [19].

Debido a la imposibilidad de acceso a la información en tiempo real (tanto los datos de intensidad de tráfico/peatonal como de cámaras de vigilancia de tráfico), se genera un código para realizar una simulación de cómo podría funcionar el sistema de forma aproximada. El escenario que se simula es similar al de la realidad en una vía donde existen dos direcciones vehiculares (izquierda y derecha) y donde existen otras dos direcciones peatonales (arriba y abajo). El lenguaje de programación empleado es Python y las herramientas principales para el desarrollo del sistema son la biblioteca Pygame para la visualización del sistema y el módulo threading para gestionar procesos concurrentes.

El objetivo principal es replicar de la manera más realista posible, el comportamiento de vehículos y peatones para ver cómo el sistema actúa mediante el semáforo ante diferentes escenarios. Para ello, se definen diferentes vehículos

y personas (variando su velocidad), se incluyen distancias de seguridad entre vehículos y se monitoriza el comportamiento de los semáforos para su posterior análisis. Es decir, la simulación no solo genera un entorno dinámico para observar el flujo de elementos del sistema, sino que también recopila datos cuantitativos que permiten evaluar la eficiencia del sistema de control de tráfico implementado.

En cuanto a la estructura general, el programa se organiza en **cuatro bloques** principales interrelacionados. Para comprender mejor la interrelación, se puede observar el Esquema 2.



Esquema 3: Desglose de los diferentes componentes del programa

En primer lugar, se define una configuración inicial para establecer los parámetros fundamentales que gobiernan la simulación, como por ejemplo las velocidades asignadas a diferentes tipos de vehículos y peatones y las coordenadas espaciales que definen las rutas de circulación. A continuación, dos clases centrales definen la lógica fundamental del sistema. Por una parte, la clase **TrafficSignal** modela el comportamiento de los semáforos, mientras que la clase **Elements**, que hereda de `pygame.sprite.Sprite`, no solo gestiona la representación visual de los vehículos/peatones sino que define también su comportamiento.

Finalmente, se definen un conjunto de tres **funciones** necesarias para el funcionamiento del sistema (**initialize()**, **generateElements()** y **simTime()**). Finalmente, la **clase** que orquesta la ejecución del sistema (**Main**) donde se ejecutan los procesos de manera concurrente, incluyendo la generación continua de vehículos y peatones (de forma aleatoria), la alternancia sincronizada de los semáforos y la visualización interactiva de toda la simulación. A continuación en la Figura se puede observar en el código de forma esquemática los tres últimos bloques mencionados con anterioridad (siguiendo el código de colores del Esquema 2 para facilitar su identificación).

```

class TrafficSignal:
>     def __init__(self, red, yellow, green): ...

class Elements(pygame.sprite.Sprite):
>     def __init__(self, lane, vehicleClass, direction_number, direction):
>         ...
>     def move(self): ...
>     def _move_right(self, is_green, is_yellow): ...
>     def _move_down(self, is_green, is_yellow): ...
>     def _move_left(self, is_green, is_yellow): ...
>     def _move_up(self, is_green, is_yellow): ...

def generateElements(): ...
def initialize(): ...
def simTime(): ...

class Main:
>     def __init__(self): ...
>     def run(self): ...
>     def _draw_traffic_lights(self): ...
>     def _draw_stats(self): ...

```

Figura 28. Muestra de Código Python

3.4. Hardware

En esta sección, inicialmente se describen brevemente los elementos que pueden englobar un sistema como el descrito anteriormente, los elementos seleccionados para la creación de un prototipo físico de sistema y, para concluir, con el estudio de la autonomía del sistema.

Se debe tener en cuenta la infraestructura que cuenta la localización donde se vaya a situar el sistema. En el caso de Madrid, está altamente capacitada para implementar este sistema sin tener que realizar modificaciones significativas al contar con sensores y cámaras ya instalados a lo largo de toda la ciudad. No obstante, es importante tener en cuenta que es posible que sean necesarias modificaciones en la infraestructura donde se quiera instalar dicho sistema.

En la Figura XXX, se puede observar un resumen de los elementos que pueden añadirse al sistema, tal y como se ha comentado en el Apartado 3.1. .

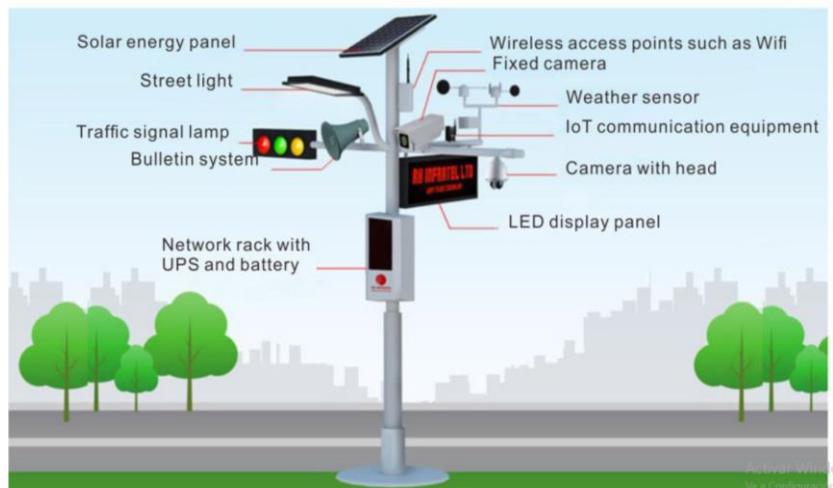


Figura 29: Elementos finales del sistema

3.4.1. Elementos para el prototipo

En base al esquema anterior, se seleccionan elementos en función de su rendimiento, calidad y precio para generar el mejor sistema posible. En la Tabla 1 se pueden encontrar un resumen de los elementos seleccionados incluyendo las fuentes de obtención de estos.

Elemento elegido	Características
------------------	-----------------

Microcontrolador: Raspberry Pi 4 Model B 8GB SC0195-9 [20]	<p>Raspberry Pi 4 cuenta con una CPU ARM Cortex-A72 de cuatro núcleos con una frecuencia de 1.5GHz (overclockable a 2.0GHz), lo que la hace significativamente más rápida que modelos anteriores de Raspberry. Dispone de hasta 8GB de RAM, lo cual es crucial para ejecutar modelos de aprendizaje automático y manejar múltiples tareas simultáneamente (por ejemplo, ejecutar una CNN, procesar datos de sensores y controlar el semáforo).</p> <p>Raspberry Pi 4 es compatible con el Módulo de Cámara V2 de Raspberry Pi o la HQ Camera, elementos esenciales para capturar imágenes para una CNN.</p> <p>Posee 40 pines de entrada/salida (GPIO), necesarios para conectar sensores (como sensores ultrasónicos, sensores IR, sensores de movimiento PIR) y para controlar los LEDs del semáforo.</p> <p>https://www.tme.eu/es/details/sc0195-9/raspberry-pi-minicomputers/raspberry-pi/raspberry-pi-4-model-b-8gb-ram/</p>
Cámara: RPI-CAMERA-V2 (Raspberry Pi Camera V2, 8MP)	Resolución de 8MP (3280x2464) https://www.tme.eu/es/details/rpi-camera-v2/accesorios-para-sistemas-embedded/raspberry-pi/raspberry-pi-camera-8mp-v2/
MICROSD: SC1645M	Capacidad de 128GB https://www.tme.eu/es/details/sc1645m/raspberry-pi-accesories/raspberry-pi/raspberry-pi-a2-class-sd-card-128gb/
Micro-HDMI to HDMI Cable: SC0358	https://www.tme.eu/es/details/sc0358/raspberry-pi-accesories/raspberry-pi/micro-hdmi-m-to-hdmi-a-m-cable-1m-white/
Fuente de alimentación: SC1408 (27W, 5V/5A, USB-C)	Especificaciones de salida: 5V DC, 5A https://www.tme.eu/es/details/sc1408/raspberry-pi-accesories/raspberry-pi/rpi-27w-usb-c-power-supply-black-eu/

Ventilador: SC0448	https://www.tme.eu/es/details/sc0448/raspberry-pi-accessories/raspberry-pi/case-fan-heatsink-for-rpi4-case/
Soporte de cámara	https://amzn.eu/d/bbwx5z0
Kit de sensores y elementos para Raspberry [21]	https://amzn.eu/d/gVfV1Cv
Semáforo LED Raspberry	https://amzn.eu/d/fjBKL84

Tabla XX: Descripción de elementos del sistema

3.4.2. Análisis de autonomía operacional

Uno de los objetivos principales del proyecto es que el sistema sea lo más autónomo y sostenible posible para evitar el colapso del tráfico en casos de situaciones de emergencia como problemas en la red eléctrica.

Teniendo en cuenta esa premisa se analizan diferentes tipos de soluciones [22]: batería grande externa, generador de diésel o gasolina, energía eólica y panel solar con batería.

Primeramente, la batería grande es descartada debido al coste elevado y a la necesidad de recarga externa. En cuanto al generador con diésel o gasolina es excluido debido a que requiere un alto coste de combustible y de mantenimiento además es ruidoso y contaminante. En lo referente a la energía eólica desechada debido a que es inconsistente, requiere una turbina y espacio y su mantenimiento es complejo y costoso.

Finalmente, el panel solar proporciona una energía gratuita e ilimitada que es favorecida por climas como el de Madrid, necesita un mantenimiento bajo, es silencioso y limpio. Además, es escalable en función de las necesidades energéticas (se pueden instalar cuantos paneles sean necesarios en función de la demanda de energía). Como desventaja es que es dependiente del clima y durante la noche depende de batería o de la red eléctrica.

- Consumo estimado del sistema del semáforo

Raspberry Pi (~3W (0.6A, 5V)) + cámara (~1.5W) + sensores (~1W): **~5.5W** (1.1A, 5V)

Luces LED del semáforo (estimado): **~9.5W** (dependiendo del modelo)
 Total de consumo de elementos: **~15W continuos**

Consumo diario de elementos: $15W \cdot 24h = 360\text{ Wh}$

- Generación solar necesaria

Se busca un panel que se ajuste a la demanda necesaria. Se selecciona un panel de 80W que según las especificaciones que se muestran en la Figura XX, genera hasta 360 Wh.

Capacidad de generación de energía media diaria.
 (Consumo de energía máximo diario de media recomendado)

~ 160-360* Wh | 1x SOLARPOWER-80W

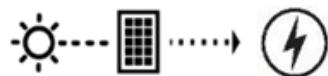


Figura XX: Especificaciones técnicas del panel 80W

Esa generación de energía depende de factores externos como las condiciones de uso, corrientes de carga y descarga, ubicación, orientación, inclinación e instalación de paneles solares y de la época del año. Suponiendo unas condiciones óptimas de generación de energía (considerando 5h de sol equivalente) se generarían 400Wh ($80W \cdot 5h$).

- Balance energético

En un día soleado de condiciones óptimas: $400\text{Wh generados} -$

$180\text{Wh (consumidos durante el día)} =$

$= 220\text{Wh de energía para almacenar}$

$220\text{Wh batería (almacenado)} - 180\text{Wh (consumidos durante la noche)} =$

$= 40\text{Wh extra}$

- Autonomía con batería

Tras una búsqueda se determina que dependiendo de las condiciones climáticas de la ciudad se recomienda usar dos tipos de baterías: de 560 Wh o de 1280 Wh. Se realizan cálculos con ambas para determinar cuál es más viable para el sistema descrito.

Autonomía (consumiendo únicamente de batería)

$$560\text{ Wh}/15\text{W} = 37.3\text{h} \approx 1.5\text{ días}$$

$$1280 \text{ Wh}/15\text{W} = 85.3\text{h} \approx 3.5 \text{ días}$$

En el caso de la ciudad de Madrid se opta por la batería de 560 Wh debido a que es una ciudad con mucho sol anualmente y para situaciones en las que no se pueda obtener energía solar, la autonomía seguiría siendo de día y medio. Para prevenir posibles inconvenientes, se hará una conexión al sistema eléctrico por si el sistema autónomo falla o si el periodo en el que no hay sol se alarga.

Teniendo en cuenta que se elige un panel de 80W y una batería de 560 Wh, se opta por seleccionar un kit ([kit panel solar](#)) cuyo datasheet se incluye en el Anexo 17. En la Figura 30, se muestra un resumen de la energía que se genera en función de la climatología, en un día nublado, durante el día el sistema podría estar funcionando prácticamente sin necesidad de la batería.

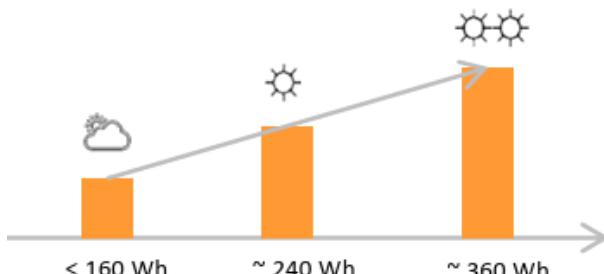


Figura 30: Gráfica de generación de energía en función de la climatología

Con el fin de ser energéticamente más eficientes, se decide optimizar el consumo energético. Por ejemplo, durante la noche, se implementa en modo de bajo consumo (reduciendo ligeramente la iluminación del semáforo). Adicionalmente, en el caso de una hora valle donde el tráfico sea bajo, dejar que el sistema funcione únicamente con los sensores, desactivando la cámara.

3.5. Problemas Técnicos y Tecnológicos

En esta sección se exponen las diferentes situaciones a las que el equipo se ha enfrentado y cómo ante dichas dificultades técnicas, se han resuelto de manera puntual y eficiente. A continuación se enlistan algunas de ellas.

Dificultades encontradas durante el ETL

Para el ETL de “Autos_Traffic”, la primera dificultad presentada fue el tamaño de cada archivo. Cada CSV mide entre 700 y 800 MB, cerca de 1'150,000,000

registros por archivo. Esto impactó en el tiempo de carga a la Base de Datos y en el tamaño de almacenamiento ya que en un principio se había propuesto un RDS en AWS. Los costos se elevaron desde un inicio por lo que, se analizó más detalladamente el conjunto de datos descubriendo que al final, para efectos del proyecto, el enfoque estaba sólo en los IDS <idelem> de la calle Gran Vía.

Así que, mediante código de python pandas, se filtraron los archivos y se generaron archivos CSV mucho más pequeños.

PROCESO ETL:

1. Python Pandas Colab
2. Para el ETL de la tabla de “PM_Trafico_Tpp_Real” Proyecto en Visual Studio para la carga de los archivos XML que actualiza la base de datos cada 5 mins. Link GitHub: Anexo 9 (pendiente subirlo)

Limpieza de Datos: La mayoría de los datos del archivo de BICICLETAS y ACCIDENTES, contenían caracteres especiales que dificultan la subida de los archivos CSV a la base de Datos.

Anexo 11: Descripción de fallas en los datos.

Bloqueo de la Suscripción de Azure por mecanismos de seguridad:

Microsoft bloquea la suscripción debido a mecanismos de seguridad automáticos que detectaron patrones inusuales en el acceso a tu base de datos SQL. Al configurar el acceso restringido únicamente a direcciones IP públicas específicas, cualquier intento de conexión desde una IP no autorizada - especialmente al correr los scripts Python que se ejecutan desde una conexión con IP dinámica puede activar alertas de seguridad.

El problema se agrava por el patrón de uso de los procesos ETL diseñados. Los scripts Python suben datos de forma masiva en intervalos irregulares (creando picos abruptos de carga), los algoritmos de detección de anomalías de Azure lo marcan como comportamiento sospechoso, similar a intentos de exfiltración de datos. Para evitar futuros bloqueos, en la segunda fase del proyecto, se planea considerar el uso de Azure Private Link para crear conexiones seguras mediante puntos de acceso privados, desde una VM con IP estática dedicada. Mejorar la autenticación implementando “Managed Identity” o “Service Principals” con certificados.

- Ejecución de queries bastante lentos:

Al requerir crecimiento la base de datos y requería mejor performance en la ejecución de queries para traer los datos. Esto impactó en el desempeño de ejecución por lo cual, se tuvo que mejorar el Pricing Tier de General Purpose - Serverless: Gen5, 4 vCores a General Purpose - Serverless: Gen5, 8 vCores.

- **Modificación de código Python:**

Se reemplazaron los queries para llenar los Data Frames por ejecución de Store Procedures (En Anexo 14 se muestra el código de su creación de los SPs).

- **Multicolinealidad**

Al llenarse los dataframes y comenzar los análisis de predicción, la matriz de correlación en el caso de “Autos_Traffic” (abajo representada) arrojan resultados importantes que se explican después de la figura 31.

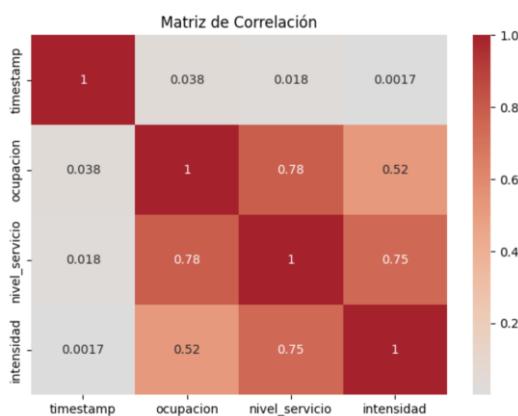


Figura 31: Comparativo de Matriz de Correlación

El resultado son correlaciones fuertes (>0.7) en ocupación y nivel_servicio: 0.78, indicando que a mayor ocupación, mayor es el nivel de servicio, nivel_servicio e intensidad (0.75) - Correlación fuerte positiva Correlaciones Moderadas (0.5-0.7): ocupación e intensidad: (0.52) Correlación moderada positiva entre ocupación e intensidad

“Timestamp” muestra correlaciones muy débiles con todas las demás variables lo que sugiere que el tiempo (timestamp) no tiene una relación lineal significativa con

otras métricas del sistema por lo que se concluye que no hay patrones temporales claros en los datos, lo que podría indicar que el sistema no sigue ciclos horarios o estacionales marcados.

Lo que llama la atención del resultado de la regresión es que el R2 es igual a uno.

```
==> MODELO DE REGRESIÓN (Intensidad) ==>
R² Score: 1.000
MAE: 0.00
```

Figura 32. Resultado de Modelo de Regresión

Las variables operacionales (ocupación, nivel de servicio, intensidad) están interrelacionadas de tal manera que pudiese ser estadísticamente correcto?

La correlaciones con el timestamp, se observan relaciones muy fuertes entre las variables principales de tráfico: ocupación, carga e intensidad presentan correlaciones altas entre sí (0.85-0.86), lo cual es lógico ya que estas métricas están intrínsecamente relacionadas y tienden a aumentar o disminuir conjuntamente cuando hay más o menos tráfico, pero en el siguiente punto se detalla algo a simple vista no se había considerado al principio.

- **Falta de acotación del sistema:**

Inicialmente, se trata de generar un sistema con todos los elementos de medición para que pueda ser un sistema general. Tras hacer varias pruebas de matriz de correlación, se observa una correlación muy baja entre variables. Mientras se busca la causa de esa falta de relación, se verifica la matriz de correlación empleando un único id de dispositivo de medición.

Tras comprobar que la relación entre las variables de entrada y la variable objetivo mejora, se determina que lo más conveniente es realizar la regresión específicamente por dispositivo. De esta manera, la naturaleza vial de la carretera no pierde su esencia y en consecuencia, no se vuelve impredecible. Es decir, el

considerar varios dispositivos que engloban patrones de funcionamiento distintos, afecta negativamente a la regresión.

- **Factores que afectan a la detección**

Tras realizar varias pruebas, se observa que existen varios factores, mayormente climatológicos (luz solar y la lluvia), que afectan a la detección de imágenes. No obstante, también se pueden dar casos de elementos de la naturaleza como ramas de árboles que pueden dificultar el desarrollo del sistema.

Por una parte, cuando los rayos del sol inciden directamente en la lente de la cámara, junto con la suciedad acumulada, genera unos reflejos y destellos que afectan a la visión computacional. A continuación, en las siguientes figuras, se muestran unos ejemplos de esa casuística.



Figura 33: Cámara de Neptuno ID 01-303



Figura 34: Cámara Gta. Ruiz Jimenez ID 01-333



Figura 35: Cámara de A.Aguilera-B.Garay
ID 07-311



Figura 36: Cámara de Callao- Gran Vía
ID 01-314

Por otra parte, cuando hay demasiada lluvia, el suelo se queda mojado y genera reflejos como si fuese un espejo. Los destellos de los elementos como farolas o vehículos deslumbran, desdibujando las siluetas y dificultando la detección de vehículos y peatones. La falta de precisión impacta de forma directa en el rendimiento del sistema. Seguidamente, en las siguientes figuras, se muestran unos ejemplos.



Figura 37: Cámara de Arturo Soria ID15-318



Figura 38: Cámara Marqués de Salamanca
ID 04-313



Figura 39: Cámara de Gran Vía-San Bernardo
ID 01-335



Figura 40: Cámara de Goya- Velázquez
ID 04-314

Adicionalmente, también se ha observado que mientras llueve, las gotas pueden tapar el objetivo de la cámara, distorsionando completamente las imágenes como se muestra en las Figuras 39 y 40.

Finalmente, en la Figura 41 o 42, se puede observar un ejemplo de cómo la naturaleza estática puede imposibilitar la detección de elementos en el sistema.



Figura 41: Cámara de Marq. de Salamanca



Figura 42: Cámara Castellana- Marq. Riscal



Figura 43: Cámara General Perón- Orense



Figura 44: Cámara Bravo Murillo - Ávila

4. Análisis y resultados

En este capítulo, se analizan los resultados obtenidos de ambos sistemas para determinar la viabilidad de las soluciones, las limitaciones de estos y las posibles líneas futuras de investigación para la mejora de estos. Adicionalmente, se hace una estimación de costes monetarios y el impacto de huella de carbono que implicaría ahorrar 5 minutos de espera en semáforos al día en una ciudad como Madrid.

4.1. Sistema basado en modelo predictivo

En esta sección, se analizan los resultados obtenidos del algoritmo desarrollado con Random Forest del device de Gran Vía '4308'. En la Figura XX, se expone la matriz de correlación, considerando "timestamp" vs "Componentes Separados"(dia, año, hora). En dichas matrices se puede observar el fenómeno descrito como multicolinealidad descrito anteriormente.

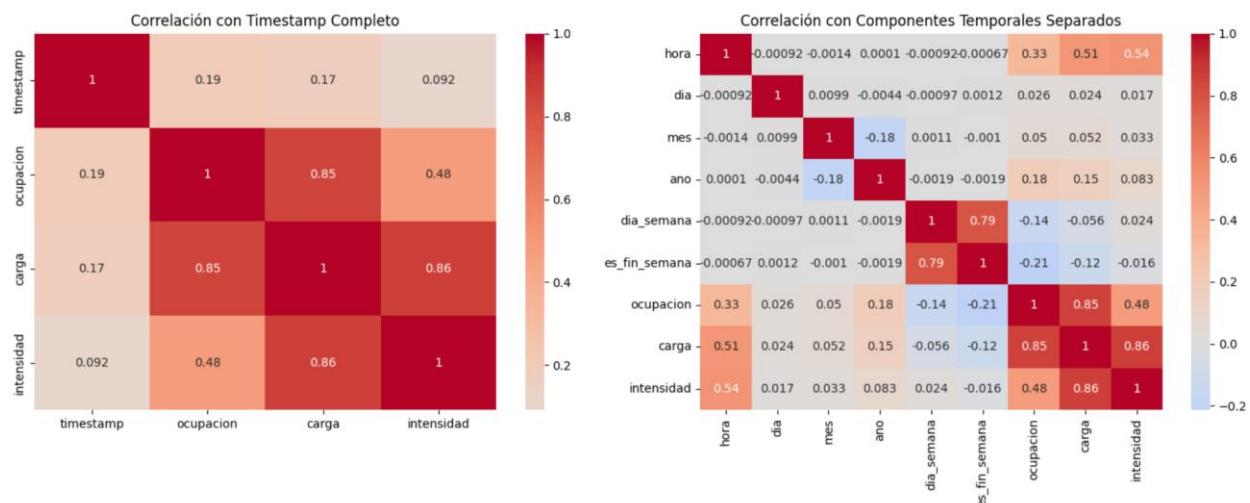


Figura 45: Comparativo Matriz de Correlaciones.

Los modelos mostraron excelente rendimiento:

- Modelo con timestamp completo: $R^2=0.988$, MAE=18.881
- Modelo con componentes separados: $R^2=0.989$, MAE=18.558

Ambos modelos explican aproximadamente el 99% de la varianza en los datos (R^2 cercano a 1), con un error absoluto alrededor de 18-19 vehículos. En el siguiente gráfico se muestra este excelente rendimiento que, aplicando hiper parámetros puede mejorar.

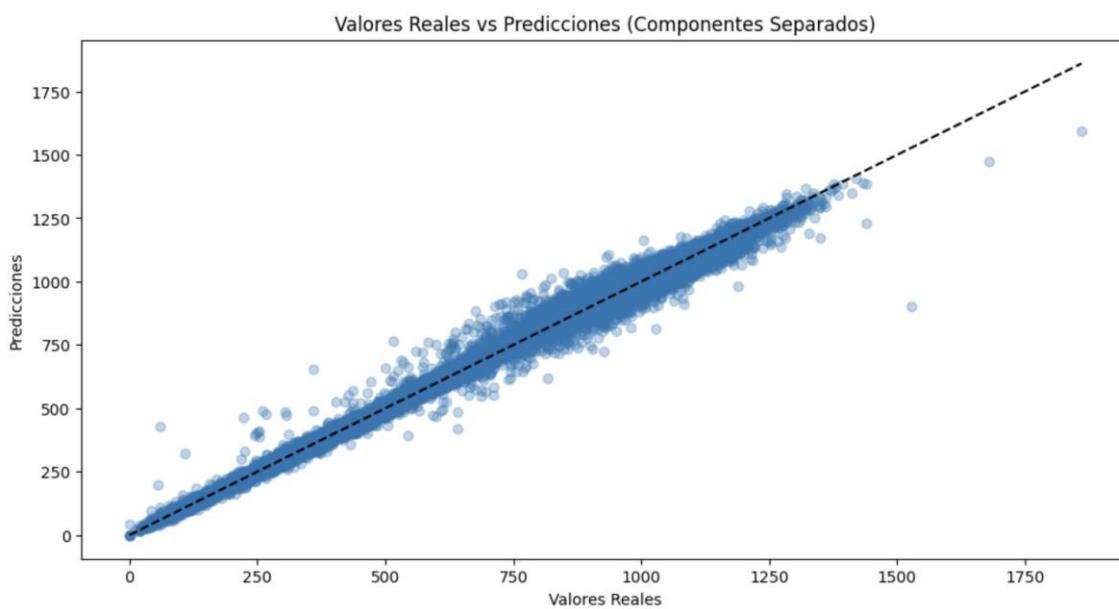


Figura 46: Rendimiento del Modelo Random Forest

El análisis revela un desempeño excepcional del modelo con características muy positivas en términos de precisión y confiabilidad. La línea diagonal punteada representa la **predicción perfecta**, donde los valores reales y predichos serían idénticos. La distribución de los puntos azules muestra una alineación muy estrecha con esta línea de referencia, lo que indica que el modelo logra predicciones muy precisas a lo largo de todo el rango de valores, desde aproximadamente 0 hasta 1600 unidades.

El modelo demuestra una capacidad predictiva consistente en diferentes niveles de intensidad de tráfico. En los valores bajos (0-400), medios (400-1000) y altos (1000-1600), la precisión se mantiene elevada sin mostrar sesgos sistemáticos hacia la sobre-estimación o sub-estimación.

La dispersión de los puntos es mínima, con muy pocos valores atípicos que se alejen significativamente de la línea diagonal. Los casos donde existe mayor variabilidad aparecen principalmente en los extremos superiores del rango (valores cercanos a 1500-1600), lo cual es común en modelos predictivos debido a la menor frecuencia de estos eventos extremos en los datos de entrenamiento.

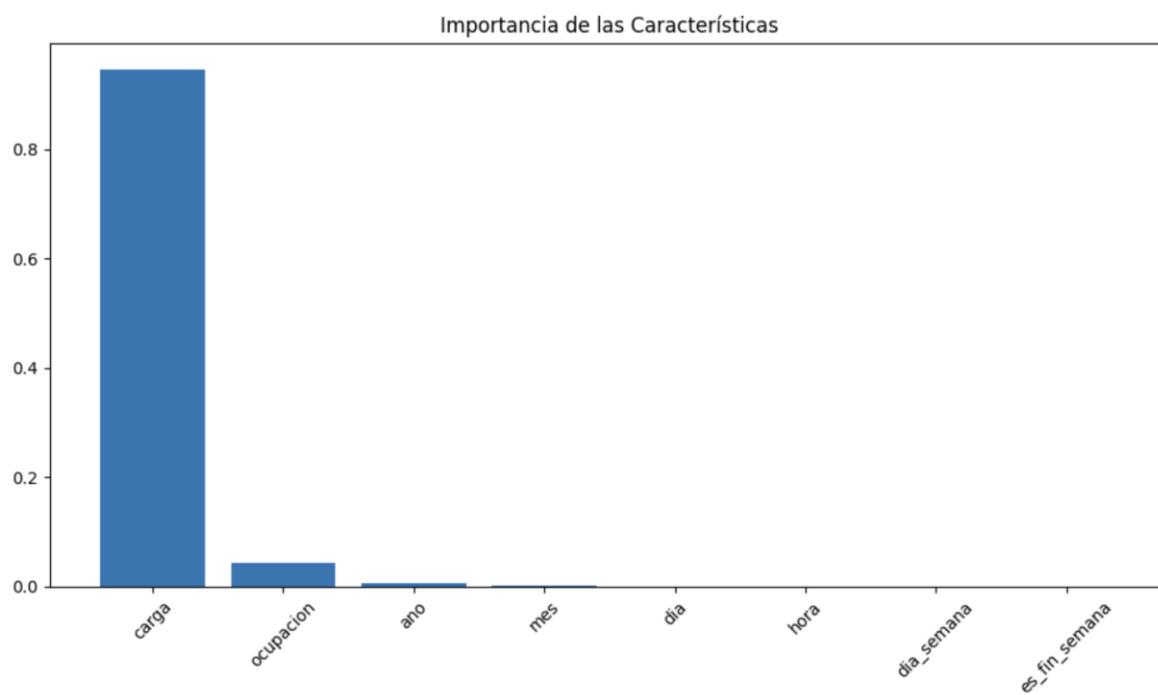


Figura 47: Importancia de las Características

La variable "carga" domina completamente el modelo con una importancia cercana a 0.95, lo que indica que es el factor más determinante para explicar el comportamiento del tráfico. Esto es lógico desde el punto de vista del análisis de tráfico, ya que la carga vehicular (porcentaje de ocupación de la vía) es una medida directa de la congestión y representa fielmente la intensidad del flujo vehicular en tiempo real. Le sigue "ocupación" con una importancia mucho menor (aproximadamente 0.05), lo cual es interesante porque aunque estas dos variables están relacionadas, "carga" tiene un poder predictivo significativamente superior.

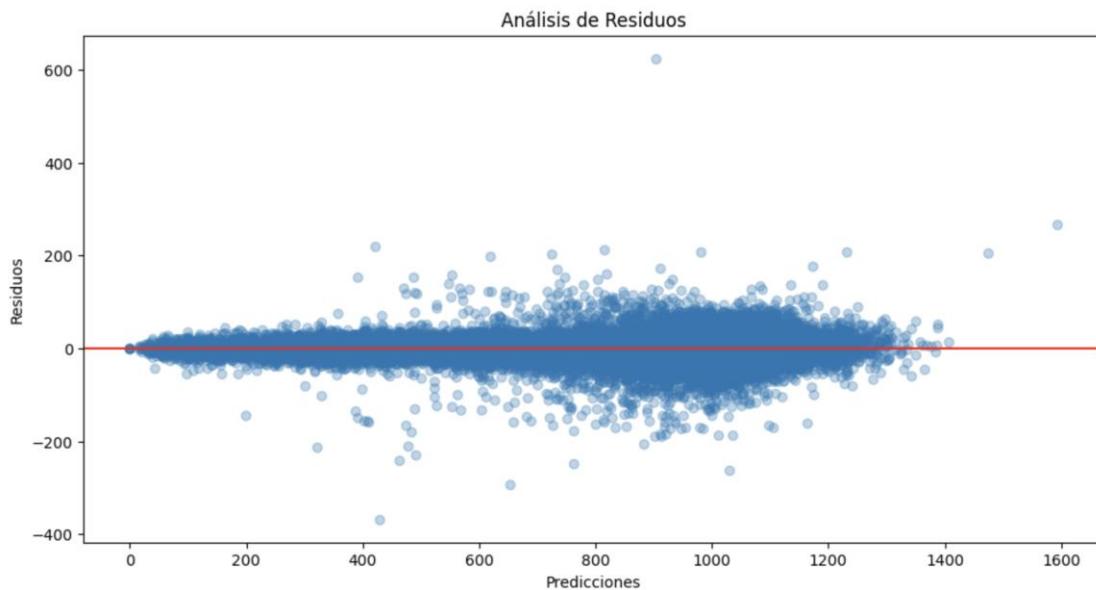


Figura 48: Análisis de Residuos

Este análisis revela características muy favorables que confirman la robustez del modelo desarrollado. La distribución de los residuos muestra una concentración muy densa alrededor de la línea roja horizontal ($\text{residuo} = 0$), lo que indica que la mayoría de las predicciones tienen errores muy pequeños. Esta concentración es particularmente notable en el rango de predicciones entre 400 y 1200 unidades.

La homocedasticidad del modelo es excelente, ya que la varianza de los residuos permanece relativamente constante a lo largo de todo el rango de predicciones. No se observan patrones de "embudo" o cambios sistemáticos en la dispersión de los errores conforme aumentan los valores predichos, lo cual es una señal muy positiva de que el modelo no sufre de heterocedasticidad.

Se identifican algunos residuos extremos, con el más notable siendo un valor cercano a +600 unidades alrededor de las 800 predicciones, pero estos casos excepcionales representan una fracción muy pequeña del total de observaciones.

Resultado del Grid Search para optimización:

```
Mejores parámetros: {'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 200}
Mejor puntuación R2: 0.9886294848821663
```

Figura 49: Resultado Grid Search

4.2. Sistema basado en visión computacional

En esta sección, se analizan los resultados obtenidos del análisis de imágenes de la cámara situada en la intersección entre Gran Vía y Clavel (01-315 ID de cámara). Adicionalmente, se analizan dos ejemplos de otras cámaras, donde se pretende argumentar la escalabilidad del sistema.

Con los datos obtenidos de las imágenes del Ayuntamiento, se realiza un análisis exhaustivo de las imágenes adquiridas para poder analizar las fortalezas y las debilidades del sistema, para conocer las limitaciones del sistema y fortalecer estas últimas en las líneas futuras de investigación. Se debe tener en cuenta que el sistema no siempre va a funcionar bajo las mismas casuísticas. En el Anexo 15 se pueden observar algunos ejemplos de situaciones anómalas que pueden suceder en la vida útil del sistema como cortes de tráfico por una carrera, manifestación o evento así como obras. Adicionalmente, se pueden encontrar diferentes ejemplos abordando distintas situaciones en el Anexo 16.

Como se puede observar en las Figuras 50 y 51 de Gran Vía, el sistema cumple con su cometido de cara a la detección de elementos en la imagen obtenida desde la cámara del Ayuntamiento.

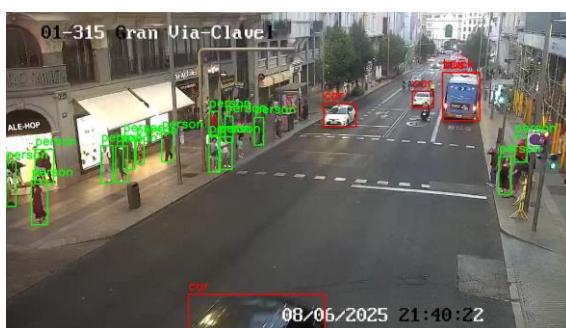


Figura 50: Cámara Gran Vía

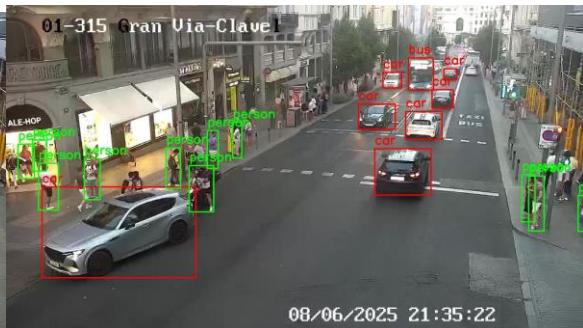


Figura 51: Cámara Gran Vía

Sin embargo, en situaciones adversas como grandes volúmenes de personas, el algoritmo no es capaz de detectar dicha magnitud como se muestra en el apartado de líneas futuras de investigación. Otro detalle que se debe corregir es que en ocasiones, los motoristas/ciclistas se consideran tanto vehículo como persona.



Figura 52: Grandes volúmenes



Figura 53: Motorista detectado como peatón

Para comprobar la escalabilidad del sistema se hacen pruebas en otras zonas de Madrid cuya casuística es diferente. En dichas pruebas, se detecta que en ocasiones, los vehículos que se encuentran a cierta distancia resultan complicados de detectar para el algoritmo. Esto se puede deber a la resolución de la imagen. Teniendo en cuenta que la imagen se trata de una matriz de píxeles, cuanto peor resolución, más complicado es para el algoritmo la detección de rasgos.



Figura 54: Detección distancia larga

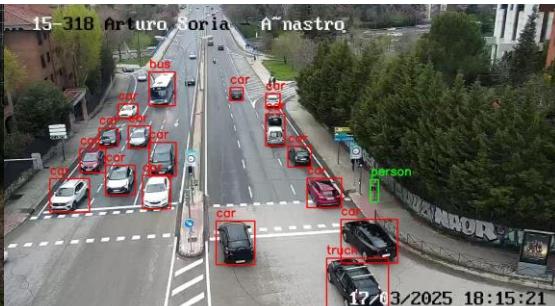
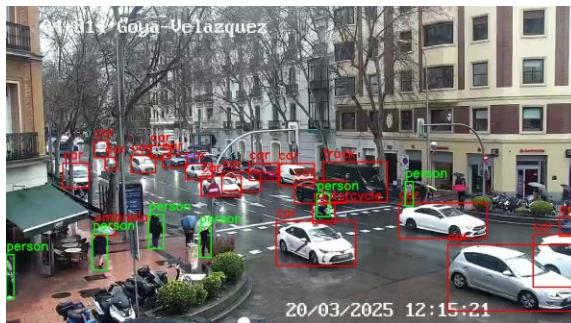


Figura 55: No detección distancia larga

Finalmente, como se muestra en las Figuras 54 y 55, la cuestión que más preocupa de cara a la escalabilidad del sistema se trata de cómo hacer que el algoritmo pueda discernir entre coches que afectan al flujo de movimiento y aquellos que están aparcados.



4.3. Análisis de costes

En esta sección se realiza una primera estimación de los costes que pueden suponer el desarrollar un proyecto de semáforos inteligentes, teniendo en cuenta diferentes posibilidades de tecnologías desde algunas más asequibles hasta la tecnología puntera en el sector.

El coste de implementar un sistema de semáforos inteligentes basado en IA y big data puede variar significativamente según la escala del proyecto y la tecnología utilizada.

Los costes iniciales incluyen:

- **Costes de instalación:** Los costes de instalación de sensores y dispositivos inteligentes pueden oscilar entre 100.000 y 500.000€.
- **Costes de software:** El desarrollo o la licencia de software de IA puede requerir una inversión de entre 20.000 y 150.000€.
- **Costes de infraestructura:** La configuración de la infraestructura necesaria para el análisis de datos puede costar entre 15.000 y 100.000€.
- **Costes de hardware:** Teniendo en cuenta los dispositivos de comunicación inalámbrica, los sistemas de control, los sistemas de alimentación de reserva puede costar entre 35.000 y 170.000€ (en función del tipo de elementos adaptados a las necesidades de cada ciudad).

Los costes de mantenimiento incluyen:

- **Tarifas de licencia de software:** Pueden oscilar entre 10.000 y 100.000 € anuales, dependiendo de los servicios que se utilicen.
- **Costes de análisis de datos:** La utilización de análisis avanzados puede suponer un coste de entre 20.000 y 150.000 € al año.
- **Salarios del personal:** Los salarios anuales de los especialistas en gestión del tráfico pueden oscilar entre 60.000 y 120.000 €.
- **Gastos de servicios públicos:** Las facturas mensuales de servicios públicos y la infraestructura de comunicaciones pueden añadir entre 10.000 y 50.000€ al presupuesto.

Pese a que la inversión inicial sea elevada debido a la necesidad de monitorización completa del sistema, la retribución se ve compensada posteriormente y no tan solo monetariamente, sino con la salud y satisfacción de los ciudadanos.

4.4. Análisis de reducción de huella de carbono

En esta sección, se analiza brevemente el impacto que puede generar al cabo de un año la reducción de espera en los semáforos de tan sólo 5 minutos.

El consumo de CO₂ de un carro frenado en un semáforo con el motor encendido depende de varios factores, como el tipo de motor, la cilindrada y el tipo de combustible. Sin embargo, podemos hacer una estimación general:

Cálculo aproximado del CO₂ emitido en 1 minuto de ralentí (marcha mínima):

- Un automóvil promedio con motor a gasolina consume entre 0.6 y 1 litro por hora en ralentí, se escoge la media de 0.8L .
- Un litro de gasolina genera aproximadamente 2.31 kg de CO₂ al quemarse.

Cálculo para 1 minuto:

- Si el coche consume 0.8 litros por hora, en 1 minuto consumirá:

- 0.8 L/h / 60min = 0.0133L/min
- Emisión de CO₂ en 1 minuto:
 - 0.0133L/min x 1min x 2.31 kg CO₂/L = 0.03 kg CO₂

Impacto en la huella de carbono:

Si un solo coche emite 0.03 kg de CO₂ en 1 minuto, teniendo en cuenta una ciudad con 1 millón de coches, si cada uno pasa 5 minutos diarios en semáforos, se podría reducir la huella de carbono en:

$$1,000,000 \times 0.03 \times 5 = 150,000 \text{ kg CO}_2/\text{día} = 150 \text{ toneladas diarias de CO}_2$$

Por tanto, teniendo en cuenta una posible reducción de 150T al día, al cabo de un año, la reducción de huella de carbono alcanzaría casi las 55.000T.

5. Propuesta de líneas futuras de investigación

En este capítulo, se pretende determinar cuáles han sido las limitaciones del sistema o ideas que se consideran interesantes de desarrollar de cara a generar un sistema más eficiente y robusto.

La principal línea de investigación consiste en un análisis de más posibilidades de algoritmos de machine learning para determinar el que mejor se ajusta al sistema descrito anteriormente. Cuanto más robusto sea el sistema, más factores se van a poder incluir como variables de medición de contaminación del aire para hacer un sistema más eficiente.

En cuanto a la detección de elementos en la imagen, se debe estudiar los diferentes filtros, resoluciones y/o algoritmos que permitan eliminar algunas de las erratas previamente descritas.

Por otra parte, se considera interesante, generar un sistema de semáforos inteligentes interrelacionados para hacer un flujo vehicular coherente a lo largo de la ciudad y probarlo en una ciudad piloto como Madrid para poder detectar posibles dificultades antes de escalarlo a otras ciudades.

Uno de los factores de mayor interés se trata de la seguridad del sistema frente a malware o ciberataques. Se cree conveniente realizar una investigación de cómo proteger a un sistema tan crucial como el de la gestión del tráfico de personas que quieran emplearlo para sus propios intereses y beneficios.

Finalmente, se cree fundamental, desarrollar una investigación enfocada puramente en la ética del algoritmo y de cómo convertir este en lo más transparente y accesible para los usuarios, así como generar una ley donde se regule las responsabilidades ante diferentes sucesos.

6. Conclusiones

Los semáforos inteligentes autónomos tienen el potencial de revolucionar la gestión del tráfico urbano y mejorar la movilidad, la seguridad y la sostenibilidad. Sin embargo, la implementación de estos sistemas presenta desafíos técnicos, legales y éticos que deben abordarse cuidadosamente, tal y como se han descrito en el Análisis de resultados y en las Líneas futuras de investigación.

La investigación futura debe centrarse en la optimización mediante distintos modelos y mejores elementos (como cámaras de mayor resolución), la integración de nuevas variables que otorguen de mayor eficiencia al sistema, la gestión de los incidentes que presenta actualmente el sistema, la seguridad y la ética del mismo.

A pesar de los desafíos, las ventajas de los semáforos inteligentes son significativas y justifican la inversión en esta tecnología. Al abordar los desafíos y aprovechar las oportunidades de investigación, podemos crear sistemas de tráfico más inteligentes, eficientes y sostenibles para las ciudades del futuro.

Ventajas

- **Reducción de la congestión:** Los semáforos inteligentes pueden reducir significativamente la congestión del tráfico, lo que se traduce en tiempos de viaje más cortos y una mayor eficiencia del combustible.
- **Mejora de la seguridad:** Los semáforos inteligentes pueden mejorar la seguridad vial al detectar y responder a posibles peligros, como peatones que cruzan la calle o vehículos que se aproximan a gran velocidad. Además, siendo un sistema autónomo, puede velar por la seguridad de los ciudadanos ante algunas adversidades como apagones generalizados.
- **Cumplimiento de los ODS con la reducción de las emisiones:** Al reducir la congestión y los tiempos de inactividad de los vehículos, los semáforos inteligentes pueden contribuir a la reducción de las emisiones de CO2 y la mejora de la calidad del aire.
- **Mayor eficiencia del transporte público:** Los semáforos inteligentes pueden dar prioridad al transporte público, lo que mejora la velocidad y la fiabilidad de los autobuses y trenes.
- **Mejora de la planificación urbana:** Los datos recopilados por los semáforos inteligentes pueden utilizarse para mejorar la planificación urbana y el desarrollo de infraestructuras.
- **Reducción de las infracciones de tráfico:** Al optimizar el flujo de tráfico y reducir la frustración de los conductores, los semáforos inteligentes pueden ayudar a disminuir la tentación de infringir las normas de tráfico, lo que contribuye a una mayor seguridad vial. Aproximadamente el 90% de los accidentes de tráfico se producen principalmente debido a infracciones de tráfico.

Desventajas

- **Costes elevados:** La implementación y el mantenimiento de los semáforos inteligentes pueden ser costosos.
- **Ciberseguridad:** Los sistemas de semáforos inteligentes son vulnerables a los ciberataques.
- **Complejidad ética:** La programación de los semáforos inteligentes para que tomen decisiones éticas en situaciones complejas es un desafío.

En definitiva, los semáforos inteligentes autónomos representan un avance transformador para las ciudades, combinando innovación tecnológica con beneficios tangibles en movilidad, seguridad y sostenibilidad. Su implementación responsable y continuo desarrollo, acercará a la humanidad a un futuro urbano más eficiente, seguro y alineado con las necesidades de la sociedad y el planeta.

Bibliografía

Antecedentes

1. Ambiente Soluciones. (s.f.). *Desarrollo: Sistema autónomo off / grid.*
<https://www.ambientesoluciones.com/portal/desarrollo-de-proyectos/desarrollo:-sistema-autonomo-off/-grid>
2. CAR Group. (2017). *MDOT international practices* [Documento PDF].
https://www.cargroup.org/wp-content/uploads/2017/02/mdot_international_practices.pdf
3. Carnegie Mellon University. (2013). *Surtrac* [Documento PDF].
https://www.ri.cmu.edu/pub_files/2013/1/13-0315.pdf
4. eHangzhou Government. (2020, 23 de octubre). *City Brain*.
https://www.ehangzhou.gov.cn/2020-10/23/c_275749.htm
5. InfoBierzo. (s.f.). *Así funcionan los nuevos semáforos con inteligencia artificial: Menos tiempo de espera y más seguros.*
https://www.infobierzo.com/tendencias/asi-funcionan-los-nuevos-semaforos-con-inteligencia-artificial-menos-tiempo-de-espera-y-mas-seguros_656401_102.html
6. Pacto Mundial Red España. (s.f.). *¿Qué puedes hacer tú? ODS.*
<https://www.pactomundial.org/que-puedes-hacer-tu/ods/>
7. PMC. (2025). Sensors on Internet of Things Systems for the Sustainable Development of Smart Cities: A Systematic Literature Review. *PubMed Central*. <https://pmc.ncbi.nlm.nih.gov/articles/PMC11014400/>

8. Wu, J., Zhang, Y., & Chen, L. (2023). HumanLight: Incentivizing ridesharing via human-centric deep reinforcement learning in traffic signal control. *arXiv preprint arXiv:2304.03697*. <https://arxiv.org/abs/2304.03697>

Metodología

9. CareerFoundry. (s.f.). *The data analysis process step by step.* <https://careerfoundry.com/en/blog/data-analytics/the-data-analysis-process-step-by-step/>

Marco Teórico

10. Centro México Digital. (s.f.). *Diferencia machine learning inteligencia artificial.* <https://centromexico.digital/diferencia-machine-learning-inteligencia-artificial/>

Desarrollo

Fuentes de Bases de Datos Tabuladas

11. Agencia Estatal de Meteorología (AEMET). (s.f.). *Acceso general AEMET.*
12. Ayuntamiento de Madrid. (s.f.). *Accidentes de tráfico de la ciudad de Madrid.*
13. Ayuntamiento de Madrid. (s.f.). *Aforo de tráfico en la ciudad de Madrid permanente del mes actual vigente que se actualiza cada 5 mins.*
14. Ayuntamiento de Madrid. (s.f.). *Aforos de peatones y bicicletas.*
15. Ayuntamiento de Madrid. (s.f.). *Aforos de tráfico en la ciudad de Madrid permanentes.*
16. Ayuntamiento de Madrid. (s.f.). *Pestañas accidentes de tráfico.*
17. Microsoft Azure. (s.f.). *SQL Azure Architecture.*

Fuentes de Base de Datos de Imágenes

18. Ayuntamiento de Madrid. (s.f.). *Informo Madrid - Tiempo real.*

<https://informo.madrid.es/#/realtime?panel=live>

Desarrollo de Código

19. Towards Data Science. (s.f.). *Traffic intersection simulation using pygame.*

<https://towardsdatascience.com/traffic-intersection-simulation-using-pygame-689d6bd7687a/>

Hardware

20. Ambiente Soluciones. (s.f.). *Desarrollo: Sistema autónomo off/grid.* <https://www.ambientesoluciones.com/portal/desarrollo-de-proyectos/desarrollo:-sistema-autonomo-off/-grid>

21. IoT Sensor & Applications Online. (2025). *Low Code IoT Platform.*

<https://sensor-online.se/>

22. TEDx Talks. (Año). *Título del video [Video].* YouTube.

<https://youtu.be/1WDagiA8fdU>

ANEXOS

ANEXO 1



Versión Agosto/2024

ESTRUCTURA DEL CONJUNTO DE DATOS

Nombre del conjunto de datos: aforos_PEATONES

Descripción: En estas tablas se recogen los datos de las estaciones permanentes de peatones.

Unidad responsable: Departamento de Planificación de la Movilidad

Frecuencia de actualización: Trimestral

Disponibilidad de los datos: 30 días después de finalizar cada trimestre.

Estructura del fichero de datos:

Número de columna	Nombre columna	Descripción	Valores esperados
1	fecha	Fecha del registro.	Formato de fecha: DD-MM-AAAA.
2	hora	Hora del registro.	Formato de hora: HH:MM.
3	identificador	Identificador único del registro.	Texto
4	peatones	Número de bicicletas registradas.	Numérico
5	device_id	Identificador único del registro.	Texto
6	Número_distrito	Número identificativo del distrito.	Numérico
7	distrito	Nombre del distrito.	Texto
8	direccion	Dirección exacta del aforo.	Texto
9	observaciones_direccion	Observaciones adicionales sobre la dirección del tráfico.	Texto
10	latitude	Latitud geográfica del registro.	Formato de número decimal (40.400000).
11	longitude	Longitud geográfica del registro.	Formato de número decimal (3.700000).

ANEXO 2



Versión Agosto/2024

ESTRUCTURA DEL CONJUNTO DE DATOS

Nombre del conjunto de datos: aforos_BICICLETAS

Descripción: En estas tablas se recogen los datos de las estaciones permanentes de bicis.

Unidad responsable: Departamento de Planificación de la Movilidad

Frecuencia de actualización: Trimestral

Disponibilidad de los datos: 30 días después de finalizar cada trimestre.

Estructura del fichero de datos:

Número de columna	Nombre columna	Descripción	Valores esperados
1	fecha	Fecha del registro.	Formato de fecha: DD-MM-AAAA.
2	hora	Hora del registro.	Formato de hora: HH:MM.
3	identificador	Identificador único del registro.	Texto
4	bicicletas	Número de bicicletas registradas.	Numérico
5	device_id	Identificador único del registro.	Texto
6	Número_distrito	Número identificativo del distrito.	Numérico
7	distrito	Nombre del distrito.	Texto
8	direccion	Dirección exacta del aforo.	Texto
9	observaciones_direccion	Observaciones adicionales sobre la dirección del tráfico.	Texto
10	latitude	Latitud geográfica del registro.	Formato de número decimal (40.400000).
11	longitude	Longitud geográfica del registro.	Formato de número decimal (3.700000).

ANEXO 3

ESTRUCTURA DEL CONJUNTO DE DATOS

Nombre del conjunto de datos: Aforos de tráfico en la ciudad de Madrid permanentes

Descripción: Este conjunto de datos ofrece los datos de aforos de tráfico registrado en las 60 estaciones permanentes disponibles en la ciudad de Madrid.

En el apartado 'Descargas' están disponibles los recursos siguientes:

- Datos de aforos desde enero de 2018, con el número de vehículos que pasan por determinados puntos de la ciudad.
- Ubicación de las estaciones permanentes
- Ubicación de las estaciones permanentes con el sentido y orientación de la medición

Unidad responsable: Dirección General de Planificación e Infraestructuras de Movilidad

Estructura del fichero de datos de aforos:

Nombre del campo	Descripción
FDIA	Fecha en la que se toma el dato
FEST	Número de estación
FSEN	<p>Sentido del tráfico. Puede tomar los siguientes valores:</p> <p>1- Sentido 1. Datos tomados de 1:00 a 12:00 1= Sentido 1. Datos tomados de 13:00 a 24:00 2- Sentido 2. Datos tomados de 1:00 a 12:00 2= Sentido 2. Datos tomados de 13:00 a 24:00</p> <p>La orientación correspondiente a los sentidos 1 y 2 se puede consultar en el fichero Ubicación de estaciones permanentes y sentidos de calles. La ausencia de valor en sentido y orientación corresponde a calles que solo tienen un sentido.</p>
HOR1.....HOR12	Hora en la que se ha tomado el dato

ANEXO 4

Este conjunto de datos esta relacionado con el de [Tráfico. Histórico de datos del tráfico desde 2013](#), indicando este último para cada punto de medida, los vehículos que pasan.

La infraestructura de puntos de medida, disponible en la ciudad de Madrid se corresponde con:

- **7.360 detectores de vehículos** con las siguientes características:
 - 71 incluyen dispositivos de lectura de matrículas
 - 158 disponen de sistemas ópticos de visión artificial con control desde el Centro de Gestión de Movilidad
 - 1.245 son específicos de vías rápidas y acceso a la ciudad
 - y el resto de los 5.886, con sistemas básicos de control de semáforos.
- **Más de 4.000 puntos de medida:**
 - 253 con sistemas para el control de velocidad, caracterización de los vehículos y doble lazo de lectura
 - 70 de ellos conforman las estaciones de toma de aforos específicas de la ciudad.
- **Sistemas de control automático** de toda la información obtenida de los detectores con el contraste continuo con patrones de comportamiento esperado, así como el seguimiento de las instrucciones marcadas por el Comité Técnico de Normalización AEN/CTN 199 “Equipamiento para la Gestión del Tráfico” y en particular las aplicaciones específicas del SC3

relativas a “Detectores y estaciones de toma de datos” y SC15 relativas a “Calidad de datos”.

En este mismo portal puede encontrar otros conjuntos de datos relacionados como son:

- Tráfico. Datos del tráfico en tiempo real. Con la información en tiempo real (actualizada cada 5 minutos)
- Tráfico. Mapa de tramas de intensidad del tráfico, con la misma información en formato KML, y con la posibilidad de verla en Google Maps o Google Earth.
- Y otros conjuntos de datos relacionados con el tráfico. Puede buscarlos, poniendo en el buscador (arriba a la derecha), la palabra "Tráfico".

ANEXO 5



1. TRÁFICO: DATOS DEL TRÁFICO EN TIEMPO REAL

La sensorización del tráfico se efectúa por medio de diversos equipamientos que permiten la realización del conteo de vehículos junto con la obtención del grado de ocupación de la vía. Estos sistemas de detección son, en su mayoría, lazos electromagnéticos que se colocan debajo del pavimento y detectan la masa metálica de los vehículos que pasan sobre ellos, siendo sistemas de gran calidad y precisión. Tienen las limitaciones que se circunscriben a la toma de datos en un único punto, además de no disponer de visión de la zona para verificar o conocer la causa de los datos que suministran. Existen alternativas que sí permiten mayores capacidades de configuración con el establecimiento de zonas de detección en vez de puntos, pero son menos precisas.

Esta información se actualiza en tiempo real, con una periodicidad de aproximadamente 5 minutos. Es el tiempo considerado necesario para proporcionar una medición real sin verse afectada por las fases semafóricas.

El objetivo principal de este conjunto de datos es proporcionar información del tráfico, en tiempo real, de la ciudad de Madrid.

En el portal también se proporcionan otros conjuntos de datos complementarios como son:

- Tráfico: Intensidad del tráfico, ubicación de los puntos de medida. Directamente relacionado con el anterior, permite la localización de los puntos de medida.
- Tráfico: Histórico de datos del tráfico desde 2013.
- Tráfico: Intensidad del tráfico. Mapa de tramas. Con la misma información en formato KML, y con la posibilidad de visualizarla en Google Maps o Google Earth.

En la ciudad de Madrid existen alrededor de 7.800 detectores de vehículos, de los que alrededor de unos 200 son sistemas ópticos de visión artificial con control desde el Centro de Gestión de la Movilidad, unos 1.400 son específicos de vías rápidas y acceso a la ciudad y, el resto, unos 6.200, son sistemas básicos de control de semáforos.

Estos 7.800 detectores conforman en la ciudad, aproximadamente, unos 4.100 puntos de medida, de los cuales 300 puntos están ubicados en M-30 y se diferencian de los del resto de la ciudad porque disponen de un sistema para el control de la velocidad, caracterización de los vehículos y doble lazo de lectura.

En todo momento se siguen las instrucciones marcadas por el Comité Técnico de Normalización AEN/CTN 199 "Equipamiento para la Gestión del Tráfico" y, en particular, las aplicaciones específicas del SC3 relativas a "Detectores y estaciones de toma de datos" y SC15 relativas a "Calidad de datos".

2. CONTENIDO DEL CONJUNTO DE DATOS

Los ficheros de los puntos de medida (ver el conjunto de datos “Tráfico: Intensidad del tráfico, ubicación de los puntos de medida”), tanto de tráfico urbano como de tráfico interurbano, se encuentran georreferenciados según la proyección European Terrestrial Reference System 1989 (ETRS89), de acuerdo con lo establecido en el Real Decreto 1071/2007, de 27 de julio, por el que se regula el sistema geodésico de referencia oficial en España.

Cabecera con la fecha y hora en que se obtuvieron los datos:

CAMPO	DESCRIPCIÓN
fecha_hora	Fecha y hora en que se obtuvieron los datos.

Para cada uno de los puntos de medida de tráfico urbano, los campos que se proporcionan son los siguientes:

Campo	Descripción
idelem	Identificador del punto de medida. Se corresponde con el campo “idelem” presente en el fichero georreferenciado y que permite su posicionamiento sobre plano e identificación del vial y sentido de la circulación.
descripcion	Denominación del punto de medida.
accesoAsociado	Código de control relacionado con el control semafórico para la modificación de los tiempos.
intensidad	Intensidad de número de vehículos por hora. Un valor negativo implica la ausencia de datos.
ocupacion	Porcentaje de tiempo que está un detector de tráfico ocupado por un vehículo. Por ejemplo, una ocupación del 50% en un periodo de 15 minutos significa que ha habido vehículos situados sobre el detector durante 7 minutos y 30 segundos. Un valor negativo implica la ausencia de datos.
carga	Parámetro de carga del vial. Representa una estimación del grado de congestión, calculado a partir de un algoritmo que usa como variables la intensidad y ocupación, con ciertos factores de corrección. Establece el grado de uso de la vía en un rango de 0 (vacía) a 100 (colapso). Un valor negativo implica la ausencia de datos.
nivelServicio	Parámetro calculado en función de la velocidad y la ocupación. Con ellos se forma una matriz de 4x4 con la que se determina cada uno

	de los niveles de servicio posibles: tráfico fluido (0), tráfico lento (1), retenciones (2) y congestión (3). Los umbrales de velocidad y los de ocupación que determinan dichos niveles de servicio varían en función del punto de medida.
intensidadSat	Intensidad de saturación de la vía en veh/hora y que se corresponde con el máximo número de vehículos que pueden pasar en el acceso a la intersección manteniéndose la fase verde del semáforo.
error	Código de control de la validez de los datos del punto de medida.
subarea	Identificador de la subárea de explotación de tráfico a la que pertenece el punto de medida.
st_x	Coordenada X UTM del centroide que representa al punto de medida en el fichero georreferenciado.
st_y	Coordenada Y UTM del centroide que representa al punto de medida en el fichero georreferenciado.

Ejemplo:

```

<pm>
  <idelem>3409</idelem>
  <descripcion>SEPÚLVEDA ENTRADA CRUCE N-S</descripcion>
  <accesoAsociado>240102</accesoAsociado>
  <intensidad>80</intensidad>
  <ocupacion>1</ocupacion>
  <carga>5</carga>
  <nivelServicio>0</nivelServicio>
  <intensidadSat>1800</intensidadSat>
  <error>N</error>
  <subarea>1718</subarea>
  <st_x>436008,175534995</st_x>
  <st_y>4472593,78531503</st_y>
</pm>

```

Para cada uno de los puntos de medida de tráfico interurbano, (vías rápidas y accesos a Madrid, M-30), se proporcionan los siguientes campos:

Campo	Descripción
idelem	Identificador del punto de medida. Se corresponde con el campo "idelem" presente en el fichero georreferenciado y que permite su posicionamiento sobre plano e identificación del vial y sentido de la circulación.
intensidad	Intensidad de número de vehículos por hora. Un valor negativo implica ausencia de datos.
ocupacion	Porcentaje de tiempo que está un detector de tráfico ocupado por un vehículo. Por ejemplo, una ocupación del 50% en un periodo de 15 minutos significa que ha habido vehículos situados sobre el detector durante 7 minutos y 30 segundos. Un valor negativo implica la ausencia de datos.
carga	Parámetro de carga del vial. Representa una estimación del grado de congestión, calculado a partir de un algoritmo que usa como variables la intensidad y ocupación, con ciertos factores de corrección. Establece el grado de uso de la vía en un rango de 0 (vacía) a 100 (colapso). Un valor negativo implica la ausencia de datos.
nivelServicio	Parámetro calculado en función de la velocidad y la ocupación. Con ellos se forma una matriz de 4x4 con la que se determina cada uno de los niveles de servicio posibles: tráfico fluido (0), tráfico lento (1), retenciones (2) y congestión (3). Los umbrales de velocidad y los de ocupación que determinan dichos niveles de servicio varían en función del punto de medida.
velocidad	Velocidad media de los vehículos detectados en el periodo de integración.
error	Código de control de la validez de los datos del punto de medida.
st_x	Coordenada X UTM del centroide que representa al punto de medida en el fichero georreferenciado.
st_y	Coordenada Y UTM del centroide que representa al punto de medida en el fichero georreferenciado.

Ejemplo:

```
<pm>
<idelem>3821</idelem>
<intensidad>900</intensidad>
<ocupacion>7</ocupacion>
<carga>54</carga>
```

```

<nivelServicio>0</nivelServicio>
<velocidad>64</velocidad>
<error>N</error>
<st_x>442925,120742158</st_x>
<st_y>4481473,97720451</st_y>
</pm>

```

3. PARÁMETROS DE TRÁFICO MÁS DESTACADOS

3.1 Carga

El parámetro carga es un indicador sintético que intenta medir el grado de saturación de una vía para la toma de decisiones sobre la duración de las fases de la regulación semafórica. En la bibliografía técnica existen múltiples fórmulas para su cálculo.

Al tratarse de un valor sintético se utiliza una fórmula para su cálculo, que depende del análisis de los ingenieros de tráfico y del análisis concreto de cada uno de los cruces a los que aplica.

En la fórmula de la carga se emplean unos parámetros que se adaptan de manera empírica para atender a las características de la vía y de su entorno.

A efectos de regulación existen dos entornos en la ciudad de Madrid:

- **Zona urbana**
- **M30.** En M30 no se utiliza el parámetro de carga, ya que no hay regulación semafórica, por lo que su valor es siempre cero.

La fórmula que se utiliza es la siguiente:

$$C = Y + (1 - Y) \times TO \times KC$$

siendo

$$Y = \frac{I}{I_s}$$

El significado de los parámetros es el siguiente:

- C: Carga del punto de medida (%)
- Y: Relación entre la intensidad (vehículos/hora) y la intensidad de saturación del punto de medida
- I: Intensidad del punto de medida (vehículos/hora)
- Is: Intensidad de saturación (capacidad máxima de la vía, vehículos/hora)
- TO: Tiempo de ocupación (tanto por 1)

- KC: Coeficiente de Carga. Por defecto se pone 1, pero es susceptible de modificarse después de un estudio de tráfico para adecuarlo a las características de la vía.

Estructura de la fórmula de carga

La fórmula de carga tiene dos términos:

- El primer término, predomina en el valor de la carga del sistema cuando estamos en una situación de carga elevada, baja la intensidad, pero el tiempo de ocupación sube.
- El segundo término, predomina cuando estamos lejos de la saturación.

Interpretación de la fórmula de carga

La interpretación de este parámetro ha de hacerse juntamente con el valor de la intensidad de vehículos de la vía:

- Una situación de elevada saturación de la vía dará un valor de carga alto y una intensidad, expresada en vehículos/hora, alta. Esto es así hasta alcanzar un máximo a partir del cual tanto carga como intensidad decrecen.
- Una situación de escaso tráfico dará un valor de carga bajo y una intensidad, expresada en vehículos/hora, baja.

Datos de carga publicados

Los valores de carga publicados son los datos brutos de carga que utilizan cada uno de los sistemas. Dado que no es un dato primario que puede variar para cada semáforo y a lo largo del tiempo con parámetros de ajuste libres para adecuarse a la realidad de cada momento, solo se publica como valor adimensional, sin justificar los coeficientes que se utilizan para su cálculo.

3.2 Nivel de servicio

El nivel de servicio es un parámetro alternativo al de carga para obtener una estimación del grado de congestión del tráfico. Si la carga es un parámetro cuantitativo calculado a partir de la intensidad (vehículos/hora) y la ocupación, el nivel de servicio utiliza la velocidad y la ocupación. En este caso, se forma una matriz de 4x4 con la que se determina cada uno de los niveles de servicio posibles: tráfico fluido, tráfico lento, retenciones y congestión. Los umbrales de velocidad (v_1, v_2, v_3, v_4) y los de ocupación (o_1, o_2, o_3, o_4) que determinan los cuatro niveles son distintos dependiendo del punto de medida.

Por sus características de indicador de carácter cualitativo y de inmediatez, se ofrece en el conjunto de datos **Tráfico. Datos del tráfico en tiempo real** codificado de forma numérica en el campo <nivelServicio> del fichero xml con periodicidad de unos 5 minutos y con código de colores en el mapa de **Informo**, pero no en el conjunto de datos **Tráfico. Histórico de datos del tráfico desde el 2013**. En este último se ofrecen mensualmente los datos de todos los puntos de medida con una frecuencia de 15 minutos, pero no el nivel de servicio por el motivo antes mencionado



En el conjunto de datos **Tráfico. Datos del tráfico en tiempo real**, el campo nivel de servicio puede adoptar los valores numéricos siguientes: fluido (0), lento (1), retenciones (2), congestión (3).

El código de colores para su representación en el mapa (**Informe**) es el siguiente: fluido (verde), lento (amarillo), retenciones (naranja), congestión (rojo).

Finalmente, hay que indicar que el conjunto de datos **Tráfico. Mapa de tramas de intensidad del tráfico** proporciona una representación complementaria, en este caso indicando la intensidad (vehículos/hora) por rangos y con un código de colores similar. También se ofrece en **Informe**, junto con los niveles de servicio.

3.3 Tiempo de ocupación

Porcentaje de tiempo que está un detector de tráfico ocupado (con un vehículo situado encima). Por ejemplo, una ocupación del 50% en un periodo de 15 minutos significa que ha habido vehículos situados sobre el detector durante 7 minutos y 30 segundos.

3.4 Intensidad

El número de vehículos por unidad de tiempo se expresa en vehículos por hora (vehículos/hora).

4. CONTACTO

Cualquier duda o comentario se puede canalizar a través del portal de datos abiertos o directamente dirigiéndose a:

Departamento de Tecnologías del Tráfico

ANEXO 6

Dirección General de Policía Municipal



vicealcaldía, portavoz,
seguridad y emergencias

| MADRID

Versión Noviembre 2023

Nombre del conjunto de datos: Accidentes de Tráfico de la Ciudad de Madrid

Descripción: Accidentes de tráfico en la Ciudad de Madrid registrados por Policía Municipal con víctimas y/o daños al patrimonio.

Unidad responsable: Dirección General de la Policía Municipal

Descripción de estructura del fichero de la aplicación SIGIT

IMPORTANTE:

- El fichero incluye un registro por persona implicada en el accidente (conductores, viajeros, peatones, testigos, etc.)
- Los ficheros de 2010 a 2018 solo registran los accidentes con heridos o con daños al patrimonio municipal.
- Los datos publicados son provisionales hasta seis meses después del año vencido.
- Actualmente no se dispone de datos por barrio.

Número de columna	Nombre columna	Descripción	Valores esperados
1	num_expediente	AAAASN>NNNNN, donde: • AAAA es el año del accidente. • S cuando se trata de un expediente con accidente.	texto
2	fecha	Fecha en formato dd/mm/aaaa	fecha
3	hora	La hora se establece en rangos horarios de 1 hora	hora
4	localizacion	calle 1 - calle 2 (cruce) o una calle	Texto
5	numero	Número de la calle, cuando tiene sentido	Número



			Número
6	cod_distrito	Código del distrito	
7	distrito	Nombre del Distrito	Texto
8	tipo_accidente*	Tipo de accidente: tipificado mas abajo	Texto
9	estado_meteorologico	Descripción climatología	Texto
10	tipo_vehiculo	tipo vehículo implicado	Texto
11	tipo_persona	Tipo persona implicada	Texto
12	rango_edad	tramo edad persona afectada	Texto
13	sexo	Puede ser :hombre Mujer o no asignado	Texto
14	cod_lesividad*	Viene tipificado mas abajo	Numero
15	lesividad	Descripción lesividad	Texto
16	coordenada_x_utm	ubicación coordenada x	Numero
17	Coordenada_y_utm	ubicación coordenada y	Numero
18	positiva_alcohol	Puede ser N o S	N o S
19	positiva_droga	Puede ser NULL o 1	NULL o 1

TIPOS DE ACCIDENTES:

Colisión doble: Accidente de tráfico ocurrido entre dos vehículos en movimiento, (colisión frontal, fronto lateral, lateral)

Colisión múltiple: Accidente de tráfico ocurrido entre más de dos vehículos en movimiento.

Alcance: Accidente que se produce cuando un vehículo circulando o detenido por las circunstancias del tráfico es golpeado en su parte posterior por otro vehículo.

Choque contra obstáculo o elemento de la vía: Accidente ocurrido entre un vehículo en movimiento con conductor y un objeto inmóvil que ocupa la vía o zona apartada de la misma, ya sea vehículo estacionado, árbol, farola, etc.

Atropello a persona: Accidente ocurrido entre un vehículo y un peatón que ocupa la calzada o que transita por aceras, refugios, paseos o zonas de la vía pública no destinada a la circulación de vehículos.



Vuelco: Accidente sufrido por un vehículo con más de dos ruedas y que por alguna circunstancia sus neumáticos pierden el contacto con la calzada quedando apoyado sobre un costado o sobre el techo.

Caída: Se agrupan todas las caídas relacionadas con el desarrollo y las circunstancias del tráfico, (motocicleta, ciclomotor, bicicleta, viajero bus, etc.,)

Otras causas: Recoge los accidentes por atropello a animal, despeñamiento, salida de la vía, y otros

LESIVIDAD:

- 01 Atención en urgencias sin posterior ingreso. - LEVE
- 02 Ingreso inferior o igual a 24 horas - LEVE
- 03 Ingreso superior a 24 horas. - GRAVE
- 04 Fallecido 24 horas - FALLECIDO
- 05 Asistencia sanitaria ambulatoria con posterioridad - LEVE
- 06 Asistencia sanitaria inmediata en centro de salud o mutua - LEVE
- 07 Asistencia sanitaria sólo en el lugar del accidente - LEVE
- 14 Sin asistencia sanitaria
- 77 Se desconoce

En blanco Sin asistencia sanitaria

ANEXO 6.1

Manual básico para la visualización de la ubicación de dispositivos (PDF)

Tráfico. Ubicación de los puntos de medida del tráfico

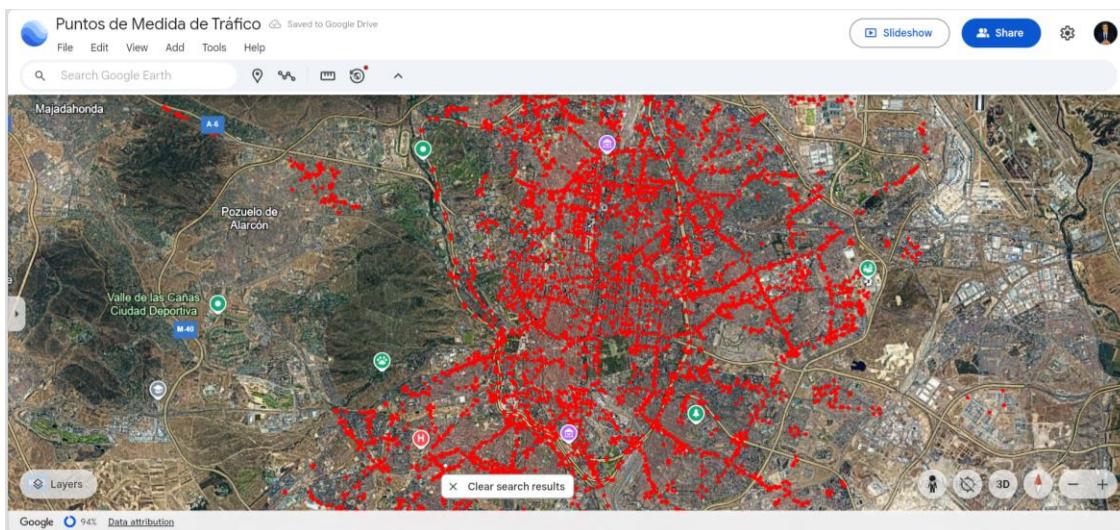
Localización de los puntos de medida del tráfico, clasificadas por tipo. Para cada elemento se detalla su ubicación tanto en sistema de coordenadas universal transversal de Mercator (en inglés Universal Transverse Mercator, UTM) y en coordenadas geográficas en WGS 84 (World Geodetic System 1984).

NOMBRE DEL CAMPO	DESCRIPCIÓN
tipo_elem	Tipo de elemento: URB urbano, M30 situado en la M30, OTHER otro tipo
distrito	Distrito
id	Identificador único numérico del punto de medida
cod_cent	Código del punto de medida
nombre	Denominación punto de medida
utm_x	Coordenada X en UTM 30N
utm_y	Coordenada Y en UTM 30N
longitud	Longitud en grados en WGS 84
latitud	Latitud en grados en WGS 84

ANEXO 6.2

Visualización interactiva de puntos de medida desde Google Earth:

https://earth.google.com/earth/d/1bi05P8DqXDO8QJk_7F1k2sA7kjf-fJPt?usp=sharing





ANEXO 7

Nombre Campo	Descripción	Tipo de Dato
fecha	fecha del dia (AAAA-MM-DD)	string
indicativo	indicativo climatológico	string
nombre	nombre (ubicación) de la estación	string
provincia	provincia de la estación	string
altitud	altitud de la estación en m sobre el nivel del mar	float
tmed	Temperatura media diaria	Float, Unidad: C
prec	Precipitación diaria de 07 a 07 Ip = inferior a 0,1 mm) (Acum = Precipitación acumulada)	Float Unidad:mm
tmin	Temperatura Mínima del día	Float, unidad:C
tmax	Temperatura Máxima del día	Float, unidad:C
horatmin	Hora y minuto de la temperatura mínima	String, unidad:UTC
horatmax	Hora y minuto de la temperatura máxima	String, unidad:UTC
DIR	Dirección de la racha máxima	Float, unidad:decenas de grado (99 = dirección variable)(88 = sin dato)
velmedia	Velocidad media del viento	Float, unidad:m/s
racha	Racha máxima del viento	float , unidad:m/s
horaracha	Hora y minuto de la racha máxima	String, unidad:UTC

sol	Insolación	Float, unidad:horas
presmax	Presión máxima al nivel de referencia de la estación	Float, unidad:hPa
horapresmax	Hora de la presión máxima (redondeada a la hora entera más próxima)	String, unidad:UTC
presmin	Presión mínima al nivel de referencia de la estación	float:hPa
horapresmin	Hora de la presión mínima (redondeada a la hora entera más próxima)	String, unidad:UTC
hrmedia	Humedad relativa media diaria	Float, unidad:%
hrmax	Humedad relativa máxima diaria	Float, unidad:%
horahrmax	Hora de la humedad relativa máxima diaria	String, unidad:UTC
hrmin	Humedad relativa mínima diaria	Float, unidad:%
horahrmin	Hora de la humedad relativa mínima diaria	string

ANEXO 8

```

import numpy as np

import pandas as pd

import os

import sys

from google.colab import drive

drive.mount('/content/drive', force_remount=True)

path1="/content/drive/MyDrive/TFM_DetecciondeTrafico_SemInt/Datos/Tra
fico_en_tpo_real/05-2025.csv" #cambia el mes según el que se suba

filename="05_2025" #cambia el mes según el que se suba

df=pd.read_csv(path1, sep=";")

Gran_Via_IdLocators = [10844, 10843, 10847, 10845, 11207, 10388,
4264, 3475, 10860, 10859, 4301, 10387, 4294, 4289, 4297, 4308,4307,
3730, 10386, 3672, 4292]

df_filtered = df[df['id'].isin(Gran_Via_IdLocators)]

from datetime import datetime

from datetime import time

#it was showing a warning, so this line is to improve the copy on
writing mode.

pd.options.mode.copy_on_write = True

df_filtered['fecha']= pd.to_datetime(df_filtered['fecha'])

#separate the fecha field for date and time columns

df_filtered['date'] = df_filtered['fecha'].dt.date

df_filtered['time'] = df_filtered['fecha'].dt.time

#saving dataframe to CSV file

rutacsv="Gran_Via_" + filename+".csv"

df_filtered.to_csv(rutacsv, index=False)

try:

```

```
df_filtered.to_csv(rutacsv, index=False) # Set index=False to avoid
saving the DataFrame index

print("Successfully saved '" + rutacsv + "'")

except Exception as e:
    print("An error occurred while saving the file: {e}")

from google.colab import files

files.download(rutacsv)
```

ANEXO 9

Links de Código GitHub

[ETL OpenDataSetClima](#)

[ETL Real Time Traffic](#)

ANEXO 10

Import flat file wizard

Step 4: Summary

1
2
3
4

Import information

Object type	Name
Server name	localhost
Database name	Traffic_Congestion
Table name	PEATONES_2019
Table schema	cbo
File to be imported	/Users/gbemba/avila/Documents/Big_Data_BI_MasterT..

Import status

```

× Microsoft.Data.SqlClient.SqlException (0x80131904): A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was
not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections. (provider: TCP Provider, error: 40 -
Could not open a connection to SQL Server)
at Microsoft.Data.SqlClient.TdsParser.ThrowExceptionAndWarning(TdsParserStateObject stateObj, Boolean callerHasConnectionLock, Boolean asyncClose)
at Microsoft.Data.SqlClient.TdsParser.Connect(SqlServerInfo serverInfo, SqlInternalConnectionTds connHandler, Boolean ignoreSniOpenTimeout, Int64 timerExpire,
SqlConnectionString connectionOptions, Boolean withFailover)
at Microsoft.Data.SqlClient.SqlInternalConnectionTds.AttemptOneLogin(ServerInfo serverInfo, String newPassword, SecureString newSecurePassword, Boolean
IgnoreSniOpenTimeout, TimeoutTimer timeout, Boolean withFailover)
at Microsoft.Data.SqlClient.SqlInternalConnectionTds.LoginNoFailover(ServerInfo serverInfo, String newPassword, SecureString newSecurePassword, Boolean
redirectedUserInstance, SqlConnectionString connectionOptions, SqlCredential credential, TimeoutTimer timeout)
at Microsoft.Data.SqlClient.SqlInternalConnectionTds.OpenLoginElist(TimeoutTimer timeout, SqlConnectionString connectionOptions, SqlCredential credential, String
newPassword, SecureString newSecurePassword, Boolean redirectedUserInstance)
at Microsoft.Data.SqlClient.SqlInternalConnectionTds..ctor(DbConnectionPoolIdentity identity, SqlConnectionString connectionOptions, SqlCredential credential, Object
providerInfo, String newPassword, SecureString newSecurePassword, Boolean redirectedUserInstance, SqlConnectionString userConnectionOptions, SessionData
reconnectSessionData, Boolean applyTransientFaultHandling, String accessToken, DbConnectionPool pool)
at Microsoft.Data.SqlClient.SqlConnectionFactory.CreateConnection(DbConnectionOptions options, DbConnectionPoolKey poolKey, Object poolGroupProviderInfo,
DbConnectionPool pool, DbConnection owningConnection, DbConnectionOptions userOptions)
at Microsoft.Data.ProviderBase.DbConnectionFactory.CreatePooledConnection(DbConnectionPool pool, DbConnection owningObject, DbConnectionOptions options,
DbConnectionPoolKey poolKey, DbConnectionOptions userOptions)
at Microsoft.Data.ProviderBase.DbConnectionPool.CreateObject(DbConnection owningObject, DbConnectionOptions userOptions, DbConnectionInternal oldConnection)
at Microsoft.Data.ProviderBase.DbConnectionPool.UserCreateRequest(DbConnection owningObject, DbConnectionOptions userOptions, DbConnectionInternal
oldConnection)
at Microsoft.Data.ProviderBase.DbConnectionPool.TryGetConnection(DbConnection owningObject, UInt32 waitForMultipleObjectsTimeout, Boolean allowCreate, Boolean
onlyOneCheckConnection, DbConnectionOptions userOptions, DbConnectionInternal& connection)
at Microsoft.Data.ProviderBase.DbConnectionPool.TryGetConnection(DbConnection owningObject, TaskCompletionSource`1 retry, DbConnectionOptions userOptions,
DbConnectionInternal& connection)
at Microsoft.Data.ProviderBase.DbConnectionFactory.TryGetConnection(DbConnection owningConnection, TaskCompletionSource`1 retry, DbConnectionOptions
userOptions, DbConnectionInternal oldConnection, DbConnectionInternal& connection)

```

Step 4: Summary

Import information

Object type	Name
Server name	svrtfmtrafficdb.database.windows.net
Database name	Traffic_Congestion
Table name	BICICLETAS_2021_
Table schema	dbo
File to be imported	/Users/gbernalavila/Documents/Big_Data_BI_Master/T...

Import status

```

X Microsoft.SqlServer.Prose.Import.BcpProcessException: Error inserting data into table.
---> System.InvalidOperationException: The given value '1:00:00 a.m.' of type String from the data source cannot be converted to type time for Column 1 [HORA].
---> System.FormatException: Failed to convert parameter value from a String to a TimeSpan.
---> System.FormatException: String '1:00:00 a.m.' was not recognized as a valid TimeSpan.
at System.Globalization.TimeSpanParse.TimeSpanResult.SetBadTimeSpanFailure()
at System.Globalization.TimeSpanParse.ProcessTerminal_HM_S_D(TimeSpanRawInfo& raw, TimeSpanStandardStyles style, TimeSpanResult& result)
at System.Globalization.TimeSpanParse.ProcessTerminalState(TimeSpanRawInfo& raw, TimeSpanStandardStyles style, TimeSpanResult& result)
at System.Globalization.TimeSpanParse.TryParseTimeSpan(ReadOnlySpan`1 input, TimeSpanStandardStyles style, IFormatProvider formatProvider, TimeSpanResult& result)
at Microsoft.Data.SqlClient.SqlParameter.CoerceValue(Object value, MetaType destinationType, Boolean& coercedToDataFeed, Boolean& typeChanged, Boolean
allowStreaming)
--- End of inner exception stack trace ---
at Microsoft.Data.SqlClient.SqlParameter.CoerceValue(Object value, MetaType destinationType, Boolean& coercedToDataFeed, Boolean& typeChanged, Boolean
allowStreaming)
at Microsoft.Data.SqlClient.SqlBulkCopy.ConvertValue(Object value, _SqlMetaData metadata, Boolean isNull, Boolean& isSqlType, Boolean& coercedToDataFeed)
--- End of inner exception stack trace ---
at Microsoft.Data.SqlClient.SqlBulkCopy.ConvertValue(Object value, _SqlMetaData metadata, Boolean isNull, Boolean& isSqlType, Boolean& coercedToDataFeed)
at Microsoft.Data.SqlClient.SqlBulkCopy.ReadWriteColumnValueAsync(Int32 col)
at Microsoft.Data.SqlClient.SqlBulkCopy.CopyColumnsAsync(Int32 col, TaskCompletionSource`1 source)
at Microsoft.Data.SqlClient.SqlBulkCopy.CopyRowsAsync(Int32 rowsSoFar, Int32 totalRows, CancellationToken cts, TaskCompletionSource`1 source)
at Microsoft.Data.SqlClient.SqlBulkCopy.CopyBatchesAsyncContinued(BulkCopySimpleResultSet internalResults, String updateBulkCommandText, CancellationToken cts,
TaskCompletionSource`1 source)
at Microsoft.Data.SqlClient.SqlBulkCopy.CopyBatchesAsync(BulkCopySimpleResultSet internalResults, String updateBulkCommandText, CancellationToken cts,
TaskCompletionSource`1 source)
at Microsoft.Data.SqlClient.SqlBulkCopy.WriteToServerInternalRestContinuedAsync(BulkCopySimpleResultSet internalResults, CancellationToken cts,
TaskCompletionSource`1 source)
at Microsoft.Data.SqlClient.SqlBulkCopy.WriteToServerInternalRestAsync(CancellationToken cts, TaskCompletionSource`1 source)
at Microsoft.Data.SqlClient.SqlBulkCopy.WriteToServerInternalAsync(CancellationToken ctoken)
at Microsoft.Data.SqlClient.SqlBulkCopy.WriteRowSourceToServerAsync(Int32 columnCount, CancellationToken ctoken)

```

A	B	C	D
31/03/21 1:00	1:00	PERM_BICI08_PM01	
31/03/21 1:00	1:00	PERM_BICI08_PM02	
31/03/21 1:00	1:00	PERM_BICI09_PM01	
31/03/21 1:00	1:00	PERM_BICI09_PM02	
31/03/21 1:00	1:00	PERM_BICI10_PM01	
31/03/21 1:00	1:00	PERM_BICI10_PM02	
31/03/21 1:00	1:00	PERM_BICI11_PM01	
31/03/21 1:00	1:00	PERM_BICI11_PM02	
31/03/21 1:00	1:00	PERM_BICI12_PM01	
31/03/21 1:00	1:00	PERM_BICI12_PM02	
31/03/21 1:00	1:00	PERM_BICI13_PM01	
31/03/21 1:00	1:00	PERM_BICI13_PM02	
31/03/21 1:00	1:00	PERM_BICI14_PM01	
31/03/21 1:00	1:00	PERM_BICI14_PM02	
31/03/21 1:00	1:00	PERM_BICI15_PM01	
31/03/21 1:00	1:00	PERM_BICI15_PM02	
31/03/21 1:00	1:00	PERM_BICI16_PM01	
31/03/21 1:00	1:00	PERM_BICI16_PM02	
31/03/21 1:00	1:00	PERM_BICI17_PM01	
31/03/21 1:00	1:00	PERM_BICI17_PM02	
31/03/21 1:00	1:00	PERM_BICI18_PM01	
31/03/21 1:00	1:00	PERM_BICI19_PM01	
31/03/21 1:00	1:00	PERM_BICI19_PM02	
31/03/21 1:00	1:00	PERM_BICI20_PM01	
31/03/21 1:00	1:00	PERM_BICI20_PM02	
01/04/21 1:00	1:00	PERM_BICI01_PM01	
01/04/21 1:00	1:00	PERM_BICI02_PM01	
01/04/21 1:00	1:00	PERM_BICI03_PM01	
01/04/21 1:00	1:00	PERM_BICI03_PM02	
01/04/21 1:00	1:00	PERM_BICI04_PM01	
01/04/21 1:00	1:00	PERM_BICI04_PM02	
01/04/21 1:00	1:00	PERM_BICI05_PM01	
01/04/21 1:00	1:00	PERM_BICI06_PM01	
01/04/21 1:00	1:00	PERM_BICI06_PM02	
01/04/21 1:00	1:00	PERM_BICI07_PM01	
01/04/21 1:00	1:00	PERM_BICI08_PM01	
01/04/21 1:00	1:00	PERM_BICI08_PM02	
01/04/21 1:00	1:00	PERM_BICI09_PM01	
01/04/21 1:00	1:00	PERM_BICI09_PM02	
01/04/21 1:00	1:00	PERM_BICI10_PM01	
01/04/21 1:00	1:00	PERM_BICI10_PM02	
01/04/21 1:00	1:00	PERM_BICI11_PM01	
01/04/21 1:00	1:00	PERM_BICI11_PM02	
01/04/21 1:00	1:00	PERM_BICI12_PM01	
01/04/21 1:00	1:00	PERM_BICI12_PM02	

ANEXO 11: Procesamiento de variables

Procesamiento - Limpieza de Datos Accidentes

Nombre de la Columna	Tipo de Datos	Descripción
idrow	bigint	
num_expediente	char(15)	NO
fecha	datetime	separar
hora	time	separar
localizacion	varchar(200)	NO Solo para limpiar Tableau
numero	varchar(50)	NO
cod_distrito	varchar(20)	NO Solo para limpiar
distrito	varchar(29)	DISTINCT: 33 Y vamos a agrupar y limpiar
tipo_accidente	varchar(35)	Son 23 sin nulos 25 con
estado_meteorologico	varchar(35)	Son 11 sin nulos. 13 con
tipo_vehiculo	varchar(60)	48 sin limpiar
tipo_persona	varchar(30)	8 conductor y pasajero, hay que limpiar

Nombre de la Columna	Tipo de Datos	Descripción
rango_edad	varchar(30)	36 Limpiar, y agrupar
sexo	varchar(15)	9 jaja
cod_lesividad	varchar(50)	NO
lesividad	varchar(80)	14 limpiar
coordenada_x_utm	varchar(30)	OK
coordenada_y_utm	varchar(30)	OK
positiva_alcohol	char(20)	9 limpiar
positiva_droga	char(20)	4 limpiar

Procesamiento:

Nombre de la Columna	Descripción	Procesamiento:
fecha	separar	
hora	separar	
distrito	DISTINCT: 33 Y vamos a agrupar y limpiar	CHAMART IN CHAMARTIN CHAMBER I CHAMBERI CHAMBERÍ Se igualaron los valores como el de ejemplo de arriba, de CHAMBER I a CHAMBERÍ.

Nombre de la Columna	Descripción	Procesamiento:
		8 REGISTROS BORRADOS CON VALOR 'NULL'
tipo_accidente	Son 23 sin nulos 25 con	
estado_meteorologico	Son 11 sin nulos. 13 con	
tipo_vehiculo	48 sin limpiar	
tipo_persona	8 conductor y pasajero, hay que limpiar	
rango_edad	36 Limpiear, y agrupar	
sexo	9 jaja	
lesividad	14 limpiar	
positiva_alcohol	9 limpiar	
positiva_droga	4 limpiar	

Un null es “null” y otro es el null del SQL

AUTOS_TRAFFIC

Nombre de la Columna	Tipo de Datos	Max
idrow	bigint	NO
id	int	NO

fecha	datetime	Separamos fecha y hora
tipo_elem	varchar(4)	NO
intensidad	varchar(10)	OK
ocupacion	varchar(10)	OK
carga	varchar(10)	OK
vmed	varchar(10)	OK
Nerror (si hubo errores en los aparatos)	char(3)	NO
periodo_integracion	char(3)	19
fechawt	date	OK
hora	time	OK

BICICLETAS

Nombre de la Columna	Tipo de Datos	Descripción
idrow	bigint	NO
fecha	datetime	OK
hora	time	OK
identificador	varchar(25)	NO

num_bicicletas	bigint	OK
device_id	varchar(25)	OK
numero_distrito	char(3)	NO
distrito	varchar(50)	Agrupar
direccion	varchar(150)	NO
observaciones_direccion	varchar(160)	NO
latitud	varchar(25)	Possible para tableau
longitud	varchar(25)	Possible para tableau

DATOS CLIMA

Variable	Explanation
fecha	Date of the observation
indicativo	Station identifier code
nombre	Name of the weather station
provincia	Province where the station is located
altitud	Altitude of the weather station in meters
tmed	Mean temperature for the day in degrees Celsius
prec	Precipitation for the day in millimeters
tmin	Minimum temperature for the day in degrees Celsius
horatmin	Time of the minimum temperature
tmax	Maximum temperature for the day in degrees Celsius
horatmax	Time of the maximum temperature

dir	Wind direction (usually in tens of degrees)
velmedia	Mean wind speed for the day
racha	Maximum wind gust for the day
horaracha	Time of the maximum wind gust
presMax	Maximum atmospheric pressure for the day
horaPresMax	Time of the maximum atmospheric pressure
presMin	Minimum atmospheric pressure for the day
horaPresMin	Time of the minimum atmospheric pressure
hrMedia	Mean relative humidity for the day
hrMax	Maximum relative humidity for the day
horaHrMax	Time of the maximum relative humidity
hrMin	Minimum relative humidity for the day
horaHrMin	Time of the minimum relative humidity

Se consume una API que existe en el web de [Open Data de Madrid](#) mediante una aplicación en Python para subir la data por trimestre, ya que sólo así lo permite la API. (Anexo 9)

Nombre de la Columna	Tipo de Datos	Descripción
idrow	bigint NULL	NO
fecha	datetime NULL	OK
indicativo	varchar 15	NO
Nombre Retiro	varchar 100	OK
Provincia (mad)	varchar 100	NO
altitud	varchar 30	OK
tmed	varchar 30	OK

prec	varchar 30	120 Nulos
tmin	varchar 30	
horatmin	varchar 30	
tmax	varchar 30	
horatmax	varchar 30	
dir	varchar 30	
velmedia	varchar 30	
racha	varchar 30	
horaracha	varchar 30	
presMax	varchar 30	OK
horaPresMax	varchar 30	OK
presMin	varchar 30	OK
horaPresMin	varchar 30	
hrMedia	varchar 30	
hrMax	varchar 30	
horaHrMax	varchar 30	
hrMin	varchar 30	

horaHrMin	varchar 30	
-----------	------------	--

PM_TRAFFIC_TPO_REAL

Results														Messages			
	idelem	descripcion	accesoAsoc	intensidad	ocupacion	carga	nivelservicio	intensidadadsat	nerror	st_x	st_y	fe					
1	3840	Jose Ortega y Ga...	0512003	140	1	13	0	830	N	441617,090594015	4475765,7692796						
2	3841	Jose Ortega y Ga...	0103004	180	1	12	0	1800	N	441706,039143845	4475769,10491597						
3	3842	Pº Recoletos N-S...	0109002	560	3	26	1	2500	N	441319,371258338	4474841,15423716						
4	3843	Pº Recoletos S-N...	0109001	920	4	19	0	3500	N	441301,63298559	4474763,7275141						
5	3844	(AFOROS) Pº Cast...	0101001	800	3	19	0	4050	N	441605,765071982	4476132,13924728						
6	3845	Pº Recoletos S-N...	0106001	820	3	18	0	3800	N	441382,968851413	4474993,8634694						
7	3846	Sagasta 20 0-E ...	0115003	999	5	47	1	1800	N	440758,420651482	4475516,99691436						
8	3847	Almagro 7 N-S - -	0117022	340	1	41	1	900	N	441885,139017928	4475582,13769423						
9	3848	(AFOROS) Genova -	0114004	1020	5	32	1	2800	N	441114,888348354	4475374,79938861						
10	3850	(AFOROS) Genova -	0106003	780	3	25	0	3250	N	441254,990156824	4475271,45945618						
11	7029	Pº Recoletos N...	0108002	820	6	19	0	3750	N	441401,492637174	4475072,29327272						
12	10034	Barbara Braganza...		380	7	26	1	765	N	441294,488284123	4474975,00966206						
13	3851	Pº Castellana N...	0106002	680	2	21	0	2700	N	441474,792659173	4475414,14427723						
14	10036	Barbara Braganza...		200	3	13	0	405	N	441134,629601492	4475019,12682102						
15	3562	(AFOROS) Pº Cast...	0103002	660	2	19	0	3100	N	441594,078438958	4475992,19154856						
16	3854	Fernando VI E-0 -	0120003	180	4	23	0	550	N	440920,452508756	4475212,93663489						
17	3852	Pº Castellana S-...	0105001	620	2	14	0	4200	N	441454,558170417	4475270,02564628						
18	7026	(MICRO) Pº RecoL...		660	3	23	0	2750	N	441404,68126178	4475071,23352072						
19	7011	(MICRO) Pl. Coló...		0	100	48	1	1500	S	441437,116590312	4475217,1832378						
20	7030	(MICRO) Pl. Coló...		540	8	36	1	1500	N	441411,753621468	4475130,01019985						
21	7032	(MICRO) Pl. Coló...		580	15	37	1	1350	N	441463,721234465	4475183,75601071						

Name	Tyyppi	DESCRIPTION
idelem	varchar(-1)	OK
descripcion	varchar(-1)	NO
accesoAsoc	varchar(-1)	NO
intensidad	varchar(-1)	OK

ocupacion	varchar(-1)	OK
carga	varchar(-1)	OK
Nivel servicio 0 Tráfico Fluido 1 Tráfico Lento 2 Retenciones 3 Congestión -1 Ausencia de datos	varchar(-1)	OK
intensidadesat	varchar(-1)	OK
nerror	varchar(-1)	NO
st_x	varchar(-1)	OK Tableau maybe
st_y	varchar(-1)	OK Tableau maybe
fechahora	varchar(-1)	Separar columnas

PEATONES

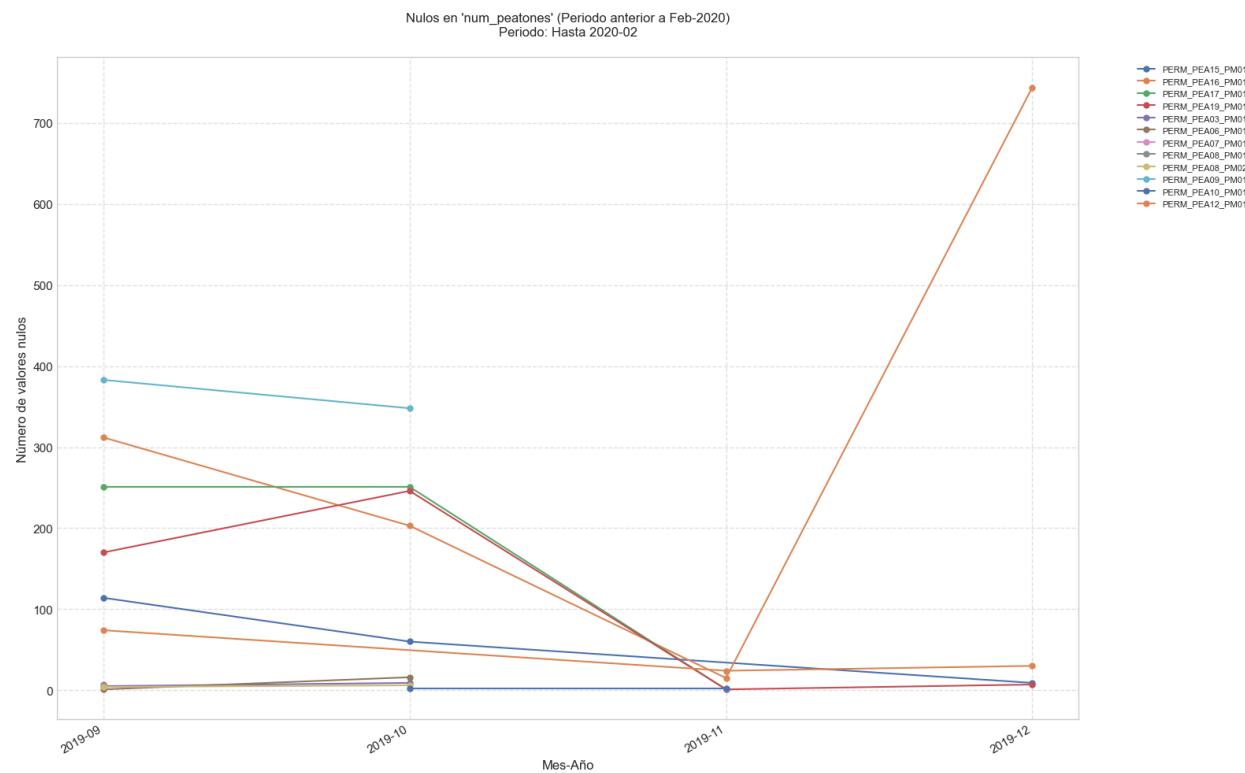
Columna	Tipo de Datos	Nueva Columna
idrow	bigint NULL	NO
fecha	datetime NULL	OK
hora	time NULL	OK
identificador	varchar 25	NO

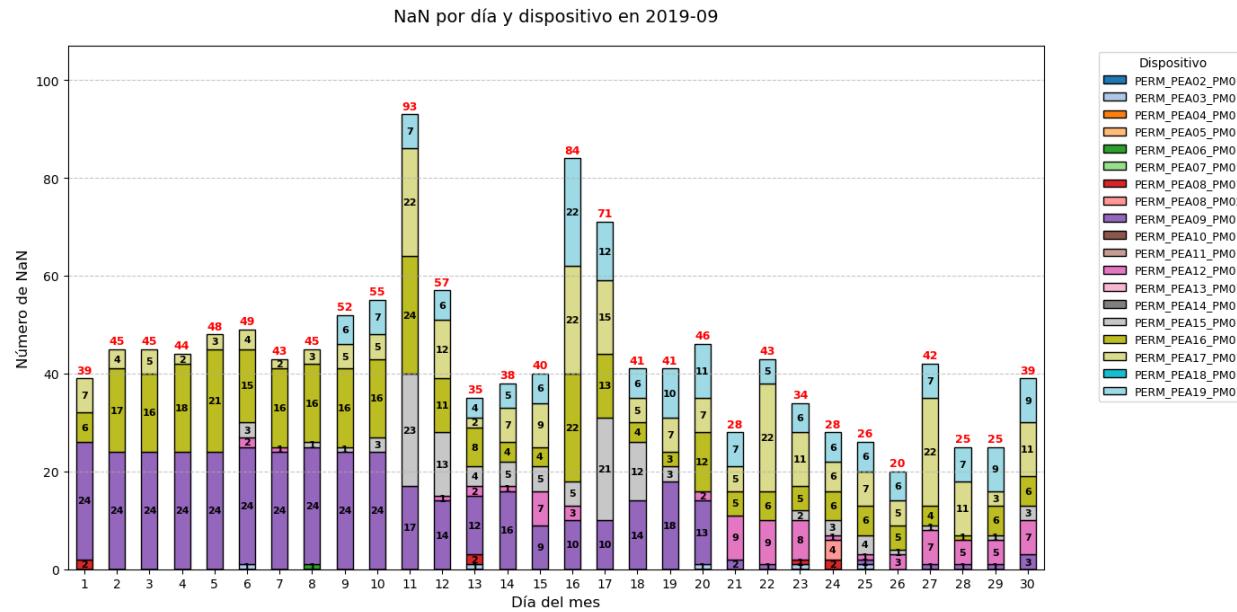
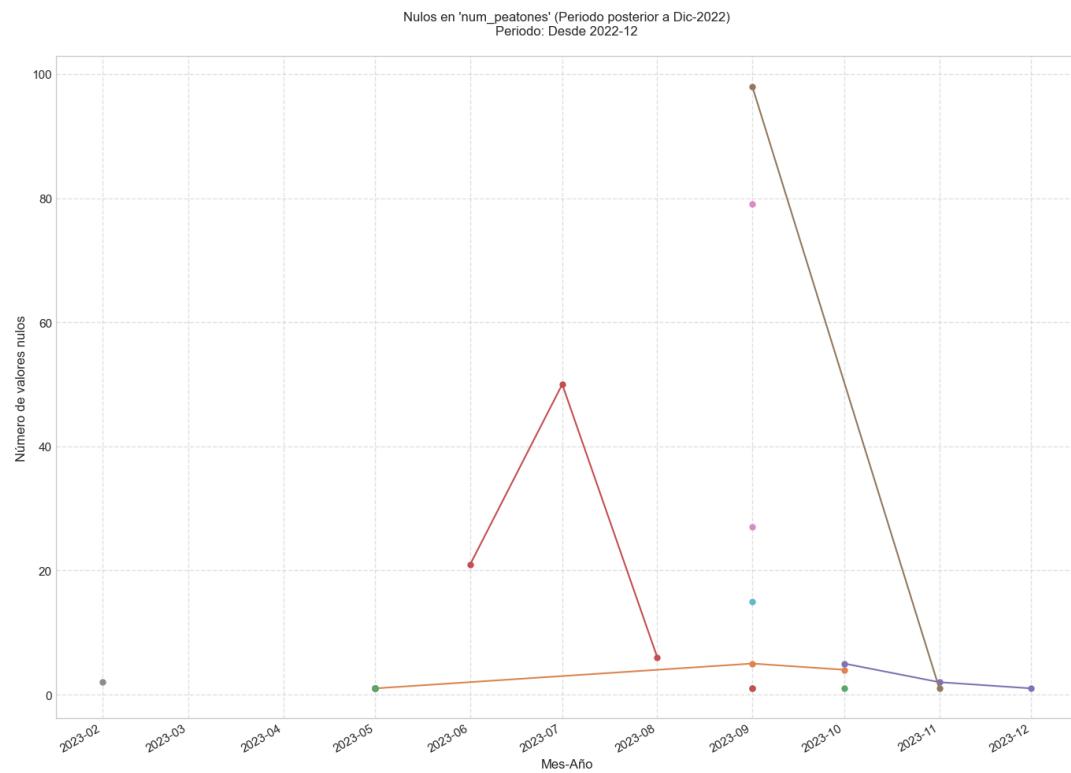
num_peatones	bigint NULL	OK
Device_id	varchar 25	OK Filtrar por gran vía
numero_distrito	char 3	OK
distrito	varchar 25	OK
direccion	varchar 100	NO
observaciones_direccion	varchar 160	OK
latitud	varchar 25	OK
longitud	varchar 25	OK

ANEXO 13: Ejemplo de análisis de nulos en peatones

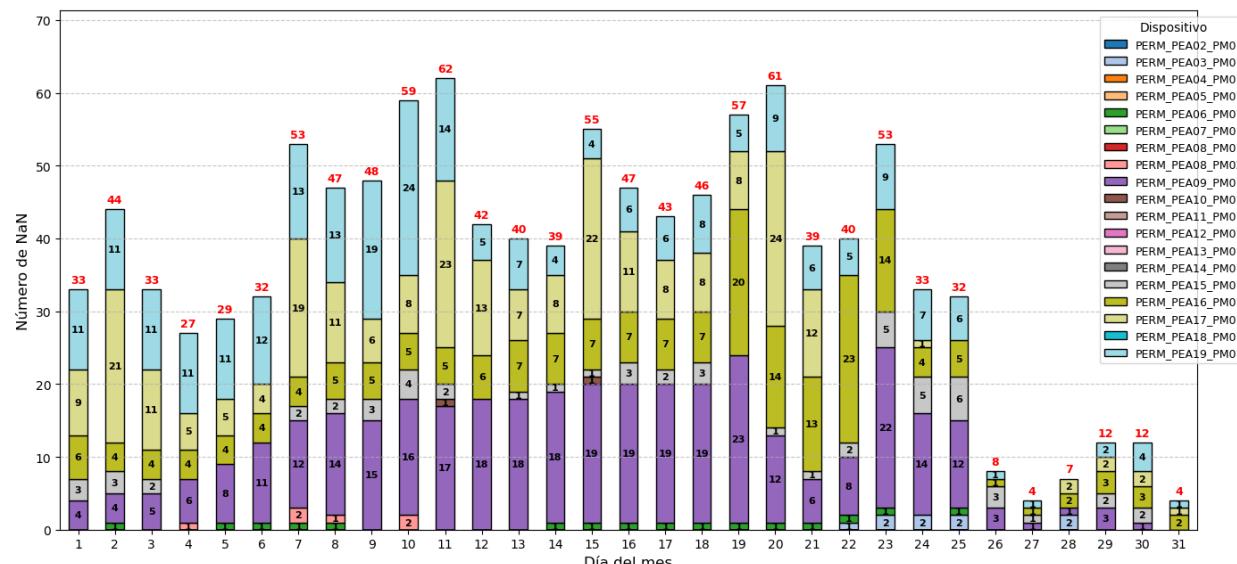
Devices	2019-09	2019-10	2019-11	2019-12	2023-02	2023-05	2023-06	2023-07	2023-08	2023-09	2023-10	2023-11	2023-12
PERM_PEA01_PM01	0	0	0	0	0	0	0	0	0	1	0	0	0
PERM_PEA02_PM01	0	0	0	0	0	1	0	0	0	0	0	0	0
PERM_PEA03_PM01	5	9	0	0	0	0	0	0	0	79	0	0	0
PERM_PEA06_PM01	1	16	0	0	0	0	0	0	0	0	0	0	0
PERM_PEA07_PM01	0	0	1	0	2	0	0	0	0	0	0	0	0
PERM_PEA08_PM01	7	0	0	0	0	0	0	0	0	0	0	0	0
PERM_PEA08_PM02	4	6	0	0	0	0	0	0	0	1	0	0	0
PERM_PEA09_PM01	383	348	0	0	0	0	0	0	0	15	0	0	0
PERM_PEA10_PM01	0	2	2	0	0	0	0	0	0	0	0	0	0
PERM_PEA11_PM01	0	0	0	0	0	1	0	0	0	0	0	0	0
PERM_PEA12_PM01	74	0	24	30	0	1	0	0	0	5	4	0	0
PERM_PEA13_PM01	0	0	0	0	0	1	0	0	0	0	0	0	0
PERM_PEA14_PM01	0	0	0	0	0	1	0	0	0	0	0	0	0
PERM_PEA15_PM01	114	60	0	9	0	0	0	0	0	0	1	0	0
PERM_PEA16_PM01	312	203	15	744	0	0	21	50	6	0	0	0	0
PERM_PEA17_PM01	251	251	1	0	0	0	0	0	0	0	0	0	0
PERM_PEA18_PM01	0	0	0	0	0	0	0	0	0	1	0	0	0
PERM_PEA19_PM01	170	246	1	7	0	0	0	0	0	0	0	0	0
PERM_PEA23_PM01	0	0	0	0	0	0	0	0	0	0	5	2	1
PERM_PEA25_PM01	0	0	0	0	0	0	0	0	0	98	0	1	0
PERM_PEA27_PM01	0	0	0	0	0	1	0	0	0	0	0	0	0
PERM_PEA28_PM01	0	0	0	0	0	0	0	0	0	27	0	0	0
TOTAL	1321	1141	44	790	2	6	21	50	6	227	10	3	1

Análisis de nulos por tiempo y dispositivo de medición. Se observa una gran cantidad de nulos inicialmente en 2019, especialmente en 5 dispositivos.

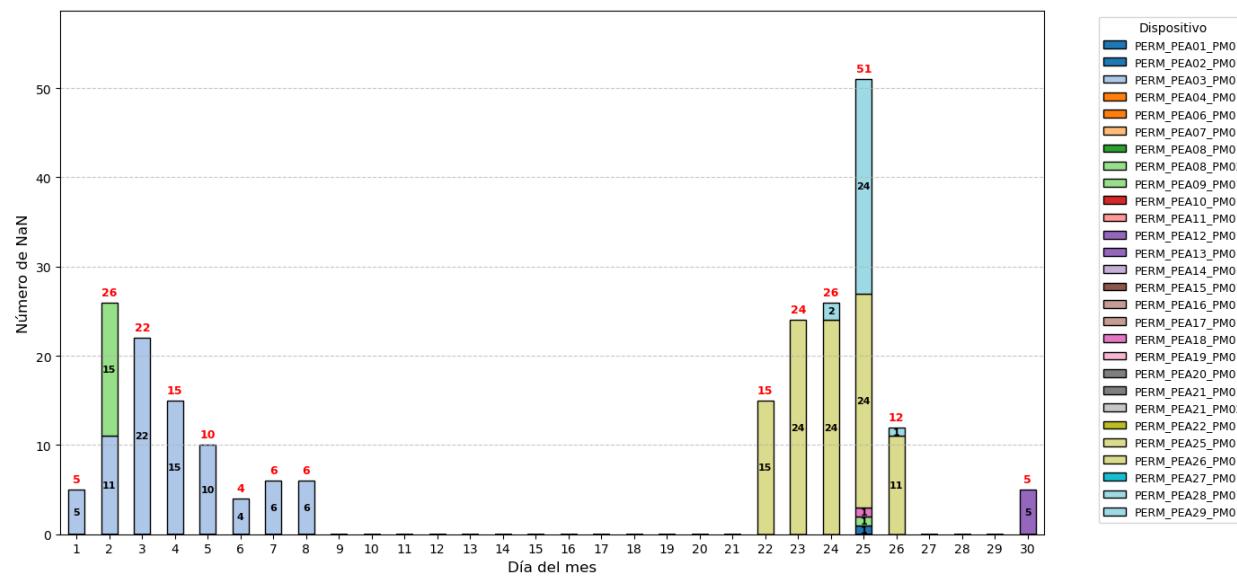




Nan por día y dispositivo en 2019-10



Nan por día y dispositivo en 2023-09



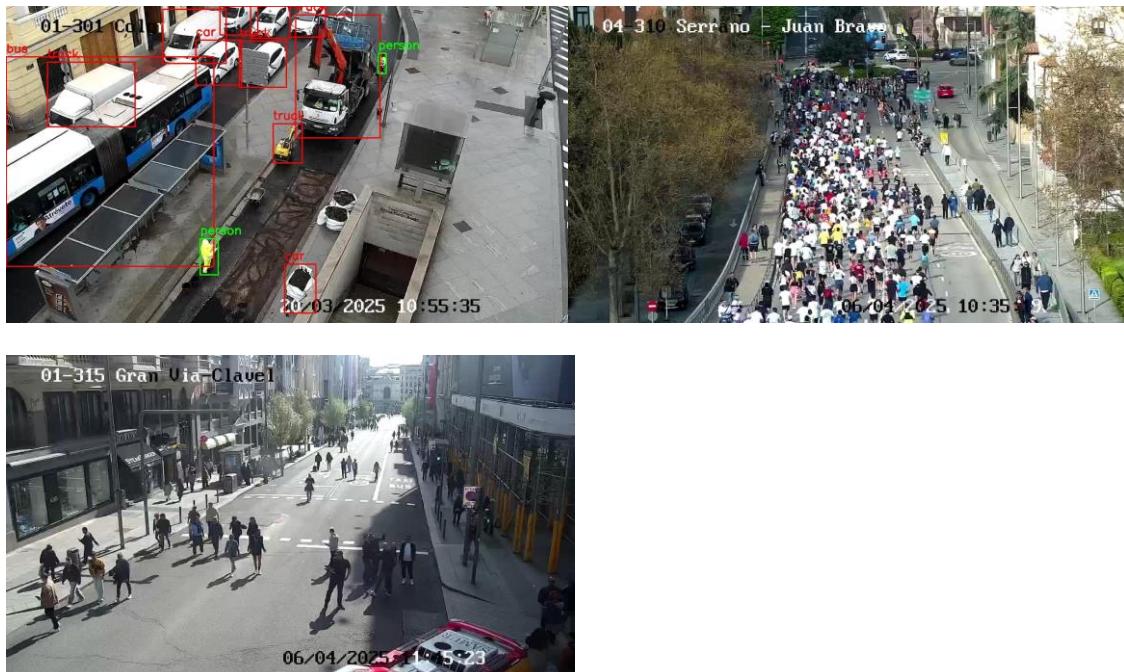
ANEXO 14: Código SQL para la creación de
Store Procedures

```

1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[SP_ACCIDENTES_CLIMA] AS
6  SELECT * FROM ACCIDENTES_CLIMA
7  GO
8
9  create procedure [dbo].[SP_AUTOS_TRAFFIC] as
10 | select * from AUTOS_TRAFFIC
11 GO
12
13 create procedure [dbo].[SP_AUTOSTRAFFIC_UBICACIONES] as
14 | select * from AUTOS_TRAFFIC_UBICACIONES
15 GO
16
17 CREATE PROCEDURE [dbo].[SP_BICICLETAS_CLIMA] AS
18 SELECT * FROM BICICLETAS_CLIMA
19 GO
20
21 CREATE PROCEDURE [dbo].[SP_PEATONES_CLIMA] AS
22 SELECT * FROM PEATONES_CLIMA
23 GO
24
25 create PROCEDURE [dbo].[SP_PM_TRAFFIC_TPO_REAL] AS
26 SELECT * FROM PM_TRAFFIC_TPO_REAL
27 GO
28
29 create PROCEDURE [dbo].[SP_UBICACIONES] AS
30 SELECT *
31 FROM UBICACIONES
32 GO
--
```

ANEXO 15: Casuísticas especiales

Obras en la calle/calle cortada por carrera manifestación u otro evento



ANEXO 16: Ejemplos de detección exitosa y fallidas

Mala calidad de las imágenes por la baja resolución de la cámara



Caso fallido con una resolución baja



Caso de éxito con resolución baja



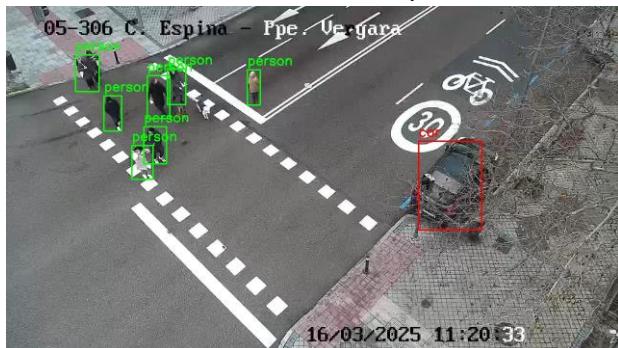
Caso fallido detección de riders



Éxito pero detecta como persona y la bicicleta



Caso de éxito de detección de peatones desde ángulo diferente de cámara

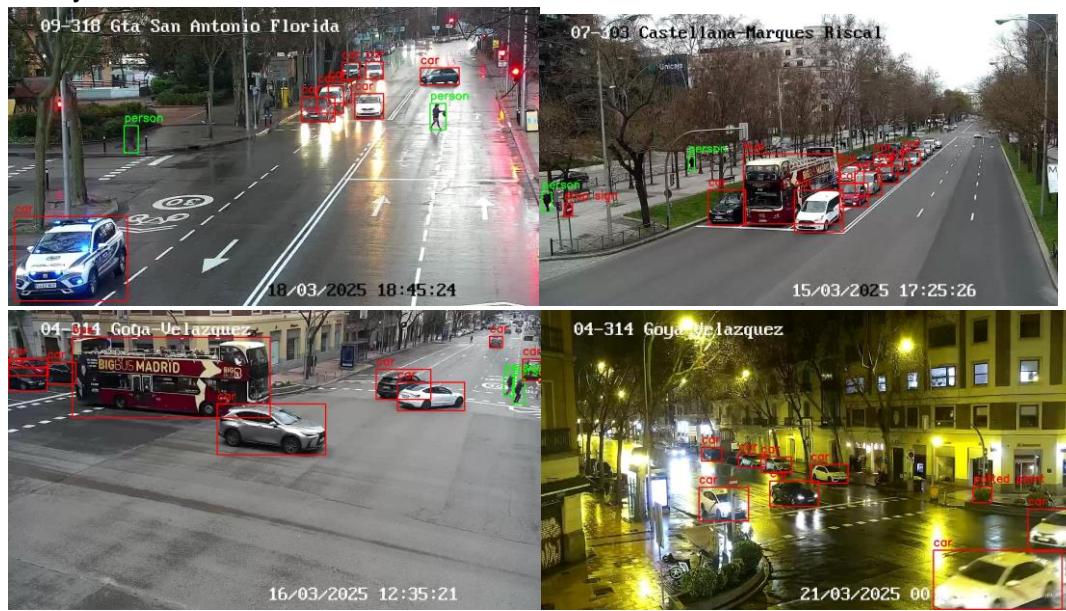


Caso de éxito de detección de elementos





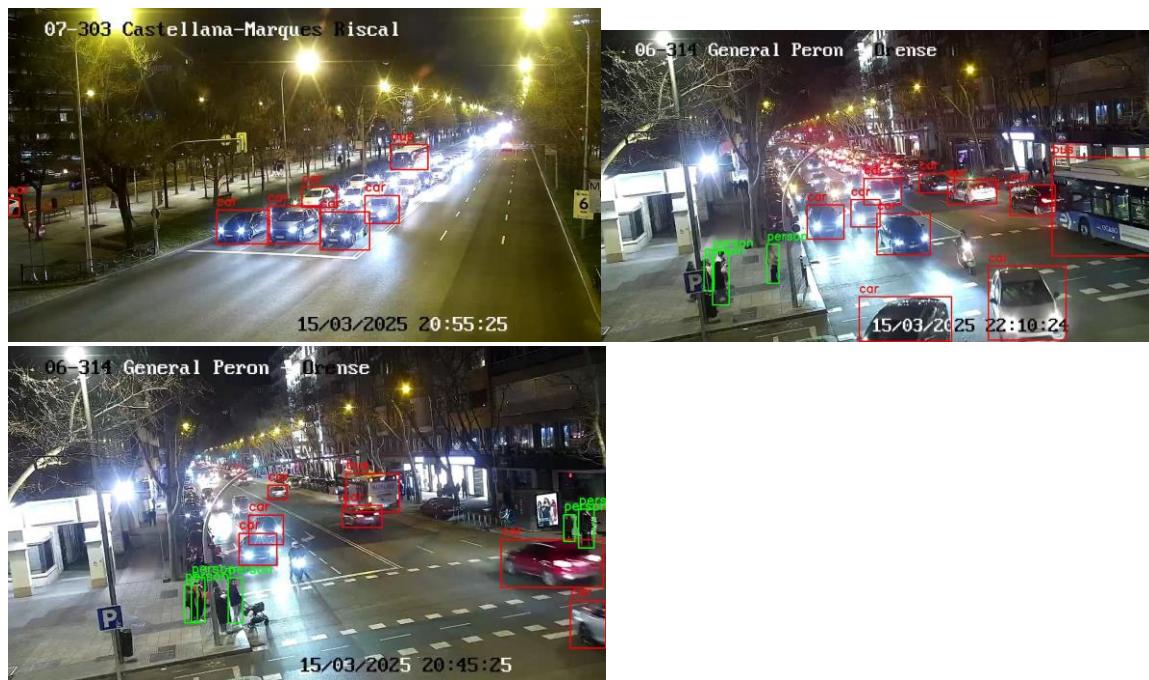
Buena detección de vehículos especiales como el de la policía, buses turísticos o plantas del ayuntamiento



Caso de éxito de detección con elementos externos (gota de agua o planta de árbol)



Casos fallidos en la detección nocturna



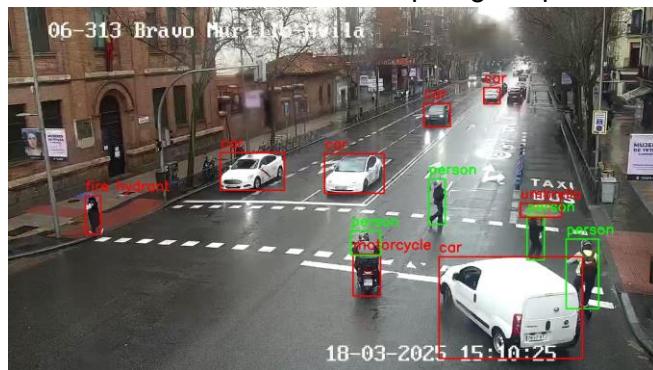
Casos de éxito en la detección nocturna



Caso de elementos aparcados y en movimiento



Caso de éxito de detección de paraguas pero detecta a persona como boca de incendio



Caso fallido de detección de moto como coche o detección de señalización de velocidad pintada en el suelo detectada como coche



Caso de detección de pantalla pero detecta edificio como pantalla también



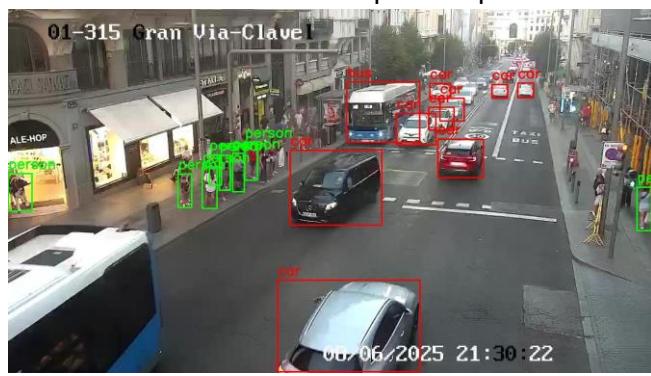
Caso especial de persona en publicidad de autobús detectado como peatón



Caso especial de carretilla detectada como camión



Detección de vaca de Alehop como peatón



Caso de detección de grandes volúmenes de personas mal detectados



Casos fallidos de detección de vehículos a distancia



ANEXO 17: Datasheet paneles solares



PACK ZERO+™ 350

code: EBAD350



PACK ZERO+™ 350 – XUNZEL



El PACK ZERO+ incluye:

- 1x panel SOLARPOWER-80W
- 1x batería SOLARX-48Ah
- 1x inversor MJ-300W-12V +USB
- 1x controlador de carga y descarga solar
- Cables y conectores
- Manuales y esquema de montaje

0,52 m²

Sistema solar fotovoltaico completo diseñado para iluminación LED y uso puntual de equipos electrónicos.

Con una potencia máxima instalada de 300W y salida USB 5V.

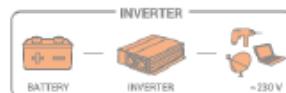
Iluminación y electricidad donde tú quieras.

Completo. Todo en uno. Listo para una fácil instalación.

Seguro, eficiente y sin mantenimiento.

Descripción

Con inversor MJ-300W-12V de onda modificada. Capaz de suministrar 300W (230VAC-50Hz) (picos de 600W). (Tiempo de uso máximo 2 h. @ 100 W - 40 min. @ 300 W). Con salida USB cargador.



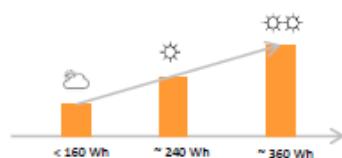
Energía disponible en la batería (cargada al 100%)

~ 460 – 580* Wh | 1x batería SOLARX-48Ah



Capacidad de generación de energía media diaria.
(Consumo de energía máximo diario de media recomendado)

~ 160-360* Wh | 1x SOLARPOWER-80W



Aplicaciones

Vivienda / Ocio
Industria / Agricultura
Seguridad / Defensa / Militar
Automatización/ Telecomunicaciones

Sistema ultra-seguro (12V)

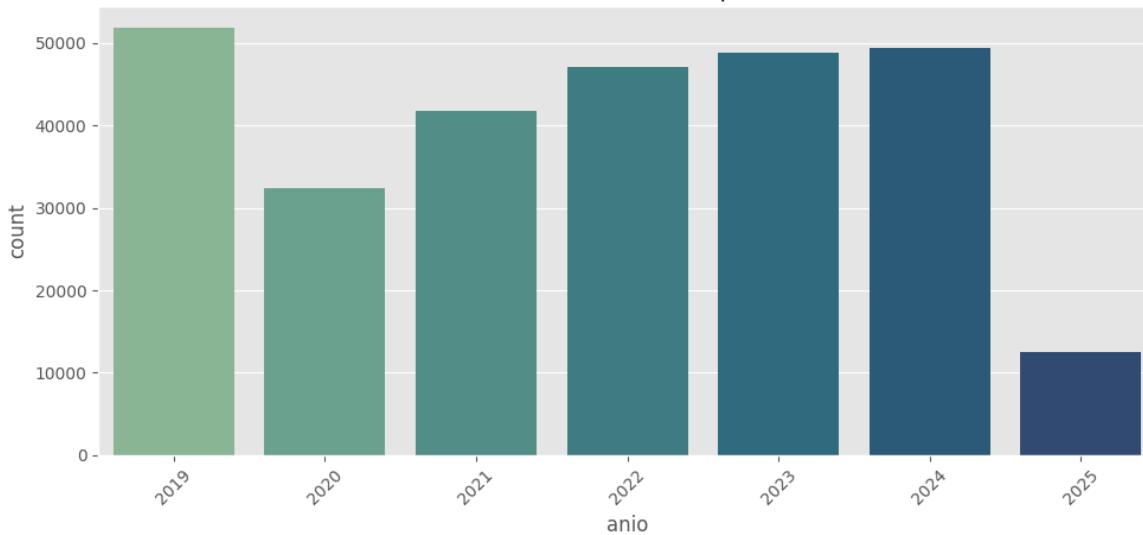
Complementos recomendados

- *LED NATURE / LED GALAXXI Series (12V)
- *Batería SOLARX-48Ah – hasta 1 adicional (conectadas en paralelo a 12V)
- *KIT de fijación de paneles solares
Tejado inclinado – 1 unidad SOLTILE1L-B - Ref. 83006354
Suelo/Cubierta plana – 1 unidad SOLSTANDWALLB- Ref. 83006348

* Los valores pueden variar considerablemente dependiendo de las condiciones de uso, corrientes de carga y descarga, ubicación, orientación, inclinación e instalación de paneles solares y de la época del año.

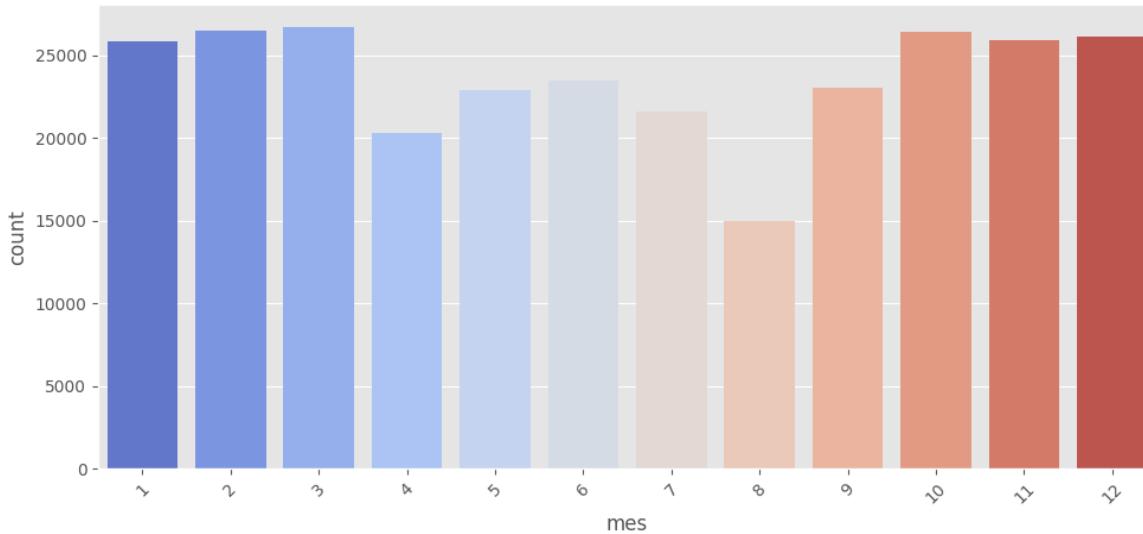
ANEXO 18: Otros gráficos importantes

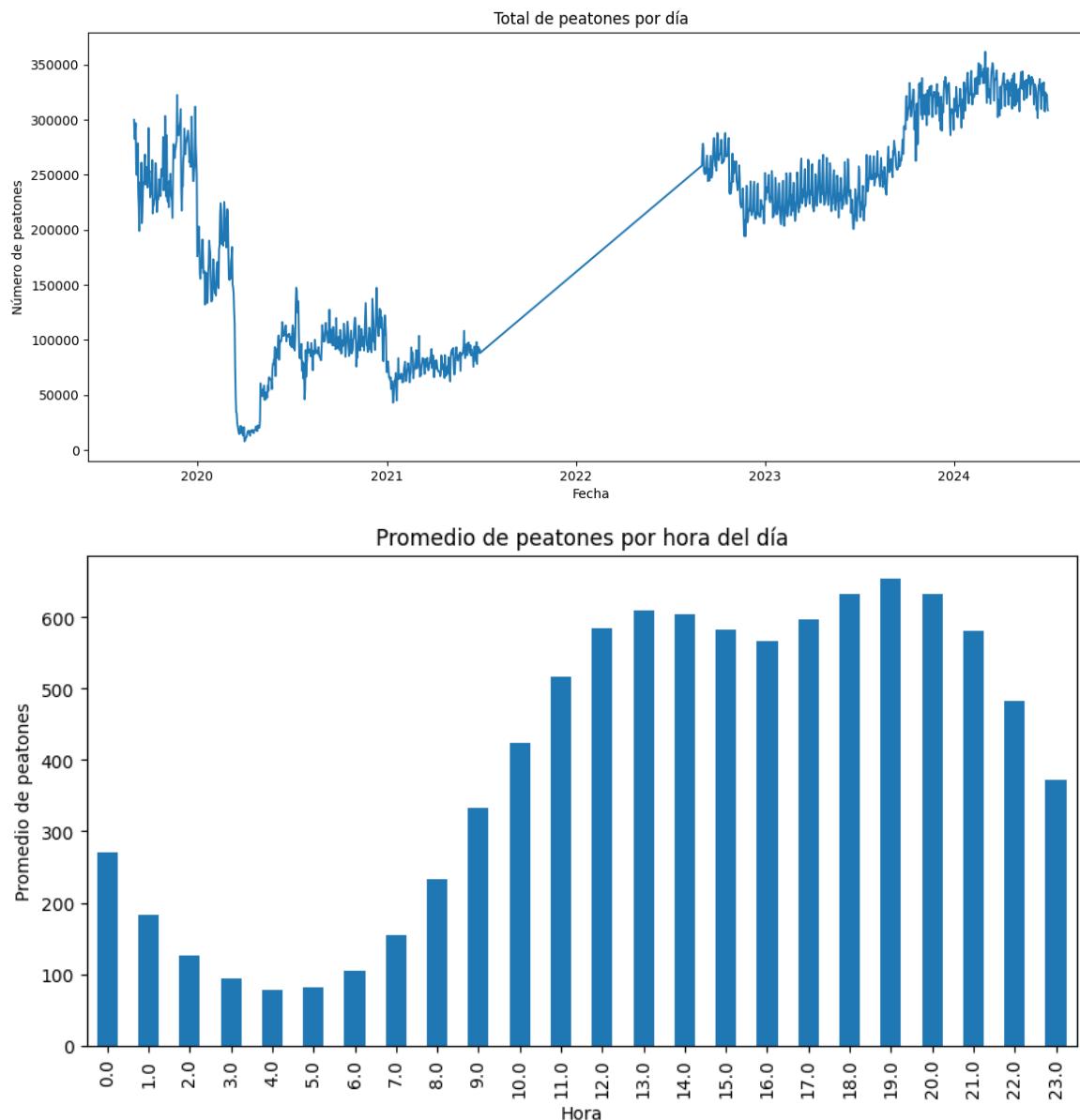
Número de accidentes por año



Accidentes por Mes

Número de accidentes por mes





Accidentes por hora (Gran Vía)

