



**Universidad
Europea**

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

**GRADO EN INGENIERÍA MATEMÁTICA APLICADA AL ANÁLISIS DE
DATOS**

PROYECTO FIN DE GRADO

**APLICACIÓN PARA EL AHORRO DE ENERGÍA Y
TIEMPO DE DESPLAZAMIENTO CON EL VEHÍCULO**

SOFÍA CORRAL CABALLERO

Dirigido por

CARLOS RAMÍREZ LIZÁN

CURSO 2021-2022

TÍTULO: Aplicación para el ahorro de energía y tiempo de desplazamiento con el vehículo

AUTOR: Sofía Corral Caballero

TITULACIÓN: Grado en ingeniería matemática aplicada al análisis de datos

DIRECTOR/ES DEL PROYECTO: Carlos Ramírez Lizán

FECHA: JUNIO de 2022

RESUMEN

Con la vuelta a la normalidad vuelven los grandes atascos en Madrid, haciendo que aumentemos el consumo de energía y, también nuestro tiempo al volante. Todo esto sumado al incremento del precio del combustible y la electricidad de los últimos meses.

Este proyecto pretende resolver el problema diseñando una aplicación para mostrar al usuario información sobre el tráfico de Madrid. En ella se han implementado datos en tiempo real haciendo uso de diferentes herramientas en la nube, idóneas para trabajar con grandes volúmenes de datos. Además, se han realizado visualizaciones enfocadas más a un análisis a largo plazo del tráfico en el municipio utilizando Power BI, una de las herramientas de visualización más utilizadas en este último período de tiempo por las empresas gracias a sus ventajas, características y facilidad de implementación. También, haciendo uso de cálculos matemáticos, se ha incorporado al producto final un modelo de predicción para ayudar a los clientes a que, tras cumplimentar una serie de variables, reciban como resultado de la consulta la hora idónea a la que circular por un punto en concreto.

Palabras clave: Análisis del tráfico en tiempo real, Modelo predictivo, Big Data, Google Cloud Platform, Power BI, Ionic

ABSTRACT

With the return to normality, the big traffic congestion in Madrid is back, increasing our energy consumption and our time behind the wheel. All this added to the increase of fuel and electricity in recent months.

This project aims to solve the problem by designing an application to show the user information about traffic in Madrid. It is implemented with real-time data using different tools in the cloud suitable for working with large volumes of data, visualizations focused more on a long-term analysis of traffic in the municipality using Power BI, one of the most used visualization tools in this last period due to its advantages, features and ease of implementation. Also, a predictive model has been incorporated into the final product to help customers, after completing a series of variables, the ideal time at which to circulate through a particular point.

Keywords: Real-time traffic analysis, Predictive model, Big Data, Google Cloud Platform, Power BI, Ionic

AGRADECIMIENTOS

A todos los que han estado a mi lado durante estos cuatro años, familiares, amigos y profesores que han hecho más ameno el camino y me han ayudado a seguir adelante siempre. En especial a mis padres que, con esfuerzo, me han permitido tener el futuro que tanto quería.

Cita - frase célebre / Dedicatoria

“El éxito es la suma de pequeños esfuerzos, repetidos día tras día”

TABLA RESUMEN

	DATOS
Nombre y apellidos:	Sofía Corral Caballero
Título del proyecto:	Aplicación para el ahorro de energía y tiempo de desplazamiento con el vehículo
Directores del proyecto:	Carlos Ramírez Lizán
El proyecto ha implementado un producto:	SI
El proyecto ha consistido en el desarrollo de una investigación o innovación:	SI (innovación)
Objetivo general del proyecto:	Crear una aplicación capaz de aportar información del tráfico de Madrid en tiempo real, de poder analizarlo a largo plazo y predecir la mejor hora para circular.

Índice

RESUMEN	3
ABSTRACT	4
TABLA RESUMEN	7
RESUMEN DEL PROYECTO	15
Contexto y justificación	15
Planteamiento del problema	15
Objetivos del proyecto	15
Resultados obtenidos	15
Estructura de la memoria	16
ANTECEDENTES / ESTADO DEL ARTE	17
Estado del arte	17
Contexto y justificación	25
Planteamiento del problema	26
OBJETIVOS	28
Objetivos generales	28
Objetivos específicos	28
Beneficios del proyecto	28
DESARROLLO DEL PROYECTO	30
Planificación del proyecto	30
Descripción de la solución, metodologías y herramientas empleadas	37
Explicación de la solución	37
Metodología	85
Herramientas empleadas	87
Recursos requeridos	88
Presupuesto	88
Viabilidad	90
Resultados del proyecto	91
DISCUSIÓN	93

CONCLUSIONES	94
Conclusiones del trabajo	94
Conclusiones personales	94
FUTURAS LÍNEAS DE TRABAJO	95
REFERENCIAS	96
ANEXOS	100

Índice de Figuras

- Figura 1. Hora y día de la semana con más accidentes (EpData, 2022)
- Figura 2. Efecto acordeón (DGT, 2018)
- Figura 3. Porcentaje de las conductas en las que se usa el móvil al volante (EpData, 2022)
- Figura 4. Evolución del tráfico en la M-30 el día 06/05/2022 (Ayuntamiento de Madrid, 2022)
- Figura 5. Evolución vuelta a la normalidad de las grandes capitales europeas (El país, 2021)
- Figura 6. Variación anual de matriculaciones (EpData, 2022)
- Figura 7. Evolución precio diario de la luz (EpData, 2022)
- Figura 8. Evolución precio de la gasolina y diésel (EpData, 2022)
- Figura 9. Ejemplo aplicación Google maps y Waze (2022)
- Figura 10. Ejemplo data frame datos extraídos.
- Figura 11. Esquema del almacenamiento de los ficheros
- Figura 12. Esquema arquitectura GCP
- Figura 13. Elección región en GCP
- Figura 14. Elección clave de almacenamiento en GCP
- Figura 15. Elección control de acceso en GCP
- Figura 16. Elección protección de los datos en GCP
- Figura 17. Resumen creación bucket en GCP
- Figura 18. Agregar un nuevo permiso en GCP Storage
- Figura 19. Tipo de permiso y parámetros a rellenar en Cloud Storage de GCP
- Figura 20. Resumen final creación bucket en GCP con el acceso público de los datos
- Figura 21. Orden carpetas en GCP
- Figura 22. Orden fichero en GCP
- Figura 23. Elección conceptos básicos en cloud function con GCP
- Figura 24. Elección tema del activador en cloud function con GCP
- Figura 25. Elección activador en cloud function con GCP
- Figura 26. Elección configuración entorno de ejecución en cloud function con GCP
- Figura 27. Elección configuración conexiones en cloud function con GCP
- Figura 28. Elección entorno de ejecución en cloud function con GCP
- Figura 29. Elección entorno de ejecución en cloud function con GCP

Figura 30. Archivos que componen la función del cloud function con GCP

Figura 31. Registros de la función en cloud function con GCP

Figura 32. Instrucciones definir el programa en cloud scheduler con GCP

Figura 33. Configuración total definición del programa en cloud scheduler con GCP

Figura 34. Configuración ejecución en cloud scheduler con GCP

Figura 35. Configuración de reintentos en cloud scheduler con GCP

Figura 36. Gráfica ejecución de la función última hora GCP

Figura 37. Gráfica ejecución de la función 12 últimas horas GCP

Figura 38. Número de ficheros almacenados por día en GCP

Figura 39. Ficheros y memoria almacenados en GCP

Figura 40. Ficheros con la unión de los datos de cada día de la semana en GCP Storage durante quince días (2-5-22/15-5-22)

Figura 41. Fichero con la unión de todos datos recogidos en GCP Storage durante quince días (2-5-22/15-5-22)

Figura 42. Diagrama de caja-bigote con la distribución de los datos de la ocupación del tráfico

Figura 43. Datos detallados de la distribución de la variable ocupación

Figura 44. Promedio de la variable ocupación de los lunes día dos y nueve de mayo

Figura 45. Promedio de la variable ocupación de los martes día tres y diez de mayo

Figura 46. Promedio de la variable ocupación de los miércoles día cuatro y once de mayo

Figura 47. Promedio de la variable ocupación de los jueves día cinco y doce de mayo

Figura 48. Promedio de la variable ocupación de los viernes día seis y trece de mayo

Figura 49. Promedio de la variable ocupación de los sábado día siete y catorce de mayo

Figura 50. Promedio de la variable ocupación de los domingos día ocho y quince de mayo

Figura 51. Gráfica de líneas que representa el promedio de la variable ocupación por días

Figura 52. Diagrama de caja-bigote con la distribución de los datos de la velocidad del tráfico

Figura 53. Datos detallados de la distribución de la variable velocidad

Figura 54. Gráfica de líneas que representa el promedio de la variable velocidad por días

Figura 55. Gráfica de líneas que representa los valores máximos recogidos de la variable velocidad por días

Figura 56. Etiqueta que muestra el número de datos recogidos con una velocidad superior a 120 km/h

Figura 57. Gráfica de líneas que representa la distribución de fecha y hora de los datos de velocidad superiores a 120 km/h

Figura 58. Gráficas de líneas que representa la relación que hay entre la variable intensidad y ocupación en función al tiempo

Figura 59. Gráficas de líneas que representa la relación que hay entre la variable intensidad y velocidad en función al tiempo

Figura 60. Imagen formulario a rellenar en la página web para calcular el modelo de predicción

Figura 61. Resultados del RMSE obtenido usando redes neuronales a corto plazo

Figura 62. Gráfico de líneas que representa el conjunto de datos de entrenamiento y las predicciones de ese conjunto y del de test

Figura 63. Brainstorming de ideas para los gráficos de visualización

Figura 64. Diseño prototipo de las vistas de visualización

Figura 65. Ejemplo visualizaciones empleadas en el análisis de datos usando Power BI

Figura 66. Imagen índice de las visualizaciones

Figura 67. Imagen cuadro de mandos datos urbanos

Figura 68. Imagen cuadro de mandos datos no urbanos

Figura 69. Imágenes visualizaciones cuadros de mandos en formato para el dispositivo móvil

Figura 70. Imágenes visualizaciones mapa implementado en la aplicación que representa la ocupación de tráfico en madrid en dos horas distintas

Figura 71. Prototipo de las pestañas de registro y login para la aplicación

Figura 72. Prototipo de las páginas que deben salir al entrar en la aplicación

Figura 73. Imagen formulario a rellenar de la aplicación para crear una cuenta

Figura 74. Imagen formulario a rellenar de la aplicación para iniciar sesión

Figura 75. Imagen integración entre las pantallas de iniciar sesión y registrarse

Figura 76. Imagen variables y un ejemplo de la base de datos creada en MySQL para almacenar los usuarios de la aplicación

Figura 77. Imágenes formularios para crear una cuenta y registrarse sin rellenar los datos

Figura 78. Imágenes formularios para crear una cuenta y registrarse con los datos rellenados y el botón activado

Figura 79. Imágenes pantalla que representa la ocupación de tráfico en Madrid en tiempo real

Figura 80. Imágenes pantalla que nos lleva a las visualizaciones creadas para el análisis de datos en la aplicación

Figura 81. Imagen índice de las visualizaciones creadas para el análisis de datos desde la aplicación de Power BI para dispositivos telefónicos

Figura 82. Imágen página con el formulario para el modelo

Figura 83. imágen de la cuenta cuenta y ddbb usada en Heroku

Figura 84. imágen de la construcción realizada en Heroku

Figura 85. Código QR que nos lleva a la pantalla de iniciar sesión de la aplicación creada como producto final

Figura 86. Logo de la aplicación y del proyecto 'TraMad'

Figura 87. Esquema de la metodología llevada a cabo durante el desarrollo del proyecto

Índice de Tablas

Tabla 1. Planificación del proyecto por actividades y semanas de cada mes

Tabla 2. Nombre del campo, descripción y ejemplo de las variables genéricas para los datos urbanos y no urbanos

Tabla 3. Nombre del campo, descripción y tipo de las variables específicas para los datos urbanos

Tabla 4. Nombre del campo, descripción y tipo de las variables específicas para los datos no urbanos

Tabla 5. Resumen de los diferentes sprints llevados a cabo en el proyecto usando metodología scrum

Tabla 6. Presupuesto total del proyecto desglosado

Capítulo 1. RESUMEN DEL PROYECTO

1.1 Contexto y justificación

Con el regreso a las oficinas vuelven los atascos kilométricos a grandes ciudades como Madrid, aumentando el estrés y el riesgo de generar un nuevo accidente. Adicionalmente, el precio del combustible y la electricidad está incrementándose a nivel global.

Con el desarrollo de la aplicación se pretende poner fin a las retenciones, reducir la contaminación y disminuir el costo del trayecto.

1.2 Planteamiento del problema

Con la realización de este trabajo se pretende crear una aplicación capaz de, en tiempo real, analizar la densidad y congestión del tráfico en Madrid e intentar ofrecer al usuario una solución para evitar los atascos haciendo uso de diferentes gráficas, mapas e incluso aplicando un modelo predictivo que nos ayude a conocer la hora o el día menos concurrido de una zona en concreto.

La idea del proyecto parte de una idea personal y, se pretende innovar y crear un producto nuevo que ayude al usuario a solventar diversos problemas que surgen al volante como accidentes, atascos o estrés.

1.3 Objetivos del proyecto

El objetivo global es generar una aplicación que aporte información en tiempo real de la circulación en Madrid y prediga la velocidad recomendada a la que debe circular el vehículo. Para ello, se aportan cuadros de mando con el análisis de datos llevado a cabo.

1.4 Resultados obtenidos

Se ha creado una aplicación en la que el consumidor pueda ver la ocupación de las carreteras en tiempo real mediante la integración de un mapa, analizar datos mediante visualizaciones implementadas desde Power BI, y predecir la hora idónea para ir a un lugar concreto del municipio de Madrid con la creación de un modelo.

Las diferentes pruebas realizadas sobre la aplicación verifican el cumplimiento de los objetivos propuestos, así como los requisitos.

Para poder obtener este resultado ha sido necesario un amplio trabajo de extracción de datos en tiempo real mediante Google Cloud Platform (GCP) y de un análisis a posteriori, con la finalidad de verificar el conjunto de datos obtenido y así asegurarnos que tanto las visualizaciones como el modelo van a darnos como resultado valores correctos.

1.5 Estructura de la memoria

La organización de la memoria se compone de 7 capítulos:

- **Capítulo 1:** Breve resumen del proyecto haciendo énfasis en el contexto o justificación, planteamiento, objetivos y resultados obtenidos.
- **Capítulo 2:** Se explica más detalladamente el contexto sobre el que parte el proyecto, las soluciones que se encuentran actualmente, ejemplos similares y el planteamiento que se va a llevar a cabo.
- **Capítulo 3:** Se detallan los objetivos generales y específicos del proyecto y los beneficios.
- **Capítulo 4:** En este apartado se explican todos los pasos seguidos durante el desarrollo del proyecto, la metodología seguida, las herramientas utilizadas a lo largo del proyecto, junto con un análisis de viabilidad, presupuesto y resultados obtenidos.
- **Capítulo 5:** Se plantea una discusión sobre los diferentes resultados obtenidos.
- **Capítulo 6:** Se exponen las conclusiones obtenidas con el desarrollo del trabajo.
- **Capítulo 7:** Se muestran posibles líneas de mejora a futuro.

Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE

2.1 Estado del arte

Tráfico y accidentes

Uno de los mayores problemas que se encuentran en grandes ciudades como Madrid viene unido de la mano del tráfico en las carreteras. Con el regreso a las oficinas, vuelven los atascos kilométricos a Madrid y, por consiguiente, aumenta el estrés de los conductores inmersos en los mismos, incrementando así, el riesgo de provocar un nuevo accidente. Según un estudio realizado por la unión sindical obrera y contrastado con un informe publicado por Inrix Global Traffic, en 2019 cada europeo perdió entre 100 y 250 horas en atascos, con un coste anual aproximado de 2.000 euros. [1]

En consonancia a los accidentes, la Dirección General de Tráfico indica ocho posibles causas de que se provoque un atasco. Una de ellas son los accidentes o averías que hacen inutilizable uno o varios carriles, pero además podemos encontrarnos otros motivos como la disminución de carriles, las incorporaciones, gloriets, conducción agresiva o con muchos frenazos, la presencia de muchos camiones o el ‘efecto mirón’ del carril contrario a un accidente. No obstante, el principal problema de los atascos es el límite de capacidad de las vías, la DGT calcula que en un carril caben alrededor de 2.200 vehículos cada hora, pero a partir de 1.700 vehículos/hora cualquier cambio de carril o maniobra provoca una disminución de la velocidad de circulación brusca. Esta capacidad límite en ocasiones no es suficiente, sobre todo si hablamos de la hora punta de grandes ciudades como Madrid. [2]

La plataforma EpData creada por Europa Press comunica, gracias a un informe publicado sobre la seguridad vial laboral en España del Real Automóvil Club de España (RACE) en el que se realizó un análisis de datos históricos que recogen información desde 2008 hasta 2016 de fuentes como el Ministerio de Trabajo, Ministerio de Sanidad, Instituto Nacional de Estadística, DGT o la Comisión Europea entre otras, que los martes entre las 8:00 y las 9:00 de la mañana es el día y hora de la semana más probable de tener un accidente de tráfico en jornada laboral, coincidiendo la hora con la franja horaria de entrada al trabajo y colegios y, por consiguiente, con uno de los intervalos en los que se localiza una mayor congestión de coches. [3]

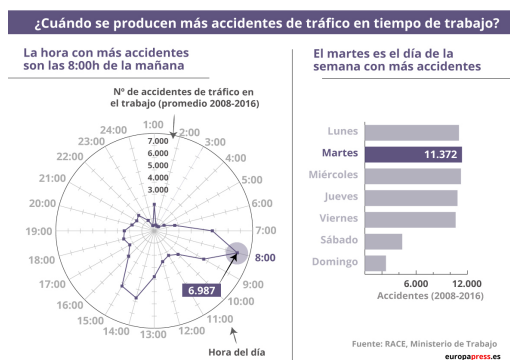


Figura 1. Hora y día de la semana con más accidentes (EpData, 2022)

Si se tiene en cuenta que existe una relación directamente proporcional entre el número de coches que circulan por la carretera con el número de accidentes que ocurren, que las horas del día con más afluencia de vehículos en Madrid es durante los días de trabajo, al fin de la jornada laboral y en las horas punta de la mañana y, que el incremento de accidentes supone la creación de atascos en las carreteras, se puede concluir que, la franja horaria con más congestión en las carreteras de Madrid coincide principalmente con la hora con más accidentes.

Una vez que el atasco ya existe, este comienza a aumentar su longitud exponencialmente, pero no sucede lo mismo en el caso de descongestionar el atasco. La Dirección General de Tráfico ve como una de las mayores causas de aumento de la congestión el llamado 'efecto acordeón', se estima que una vez arranque el primer vehículo, el siguiente lo hará 1 segundo más tarde y así sucesivamente. Si nos encontramos en un atasco de 16 kilómetros con aproximadamente 2.000 coches colocados en línea, la diferencia de tiempo desde el primer coche al último está en torno a la media hora. [4]



Figura 2. Efecto acordeón (DGT, 2018)

Por otro lado, el uso del teléfono móvil al volante es una de las mayores causas de accidente que hay en la actualidad, según una encuesta realizada en 2019 por BP, Castrol y RACE. en su campaña de seguridad vial, uno de los objetivos principales es intentar evitar las distracciones al volante, enfocando esto en los dispositivos telefónicos, donde se dice que un 25% de los conductores reconoce usar el móvil mientras van conduciendo para consultar el GPS, siendo esta la mayor causa de uso de los teléfonos, por debajo de Whatsapp con un 19 % o hablar sin hacer uso del manos libres con un 14%. [5]

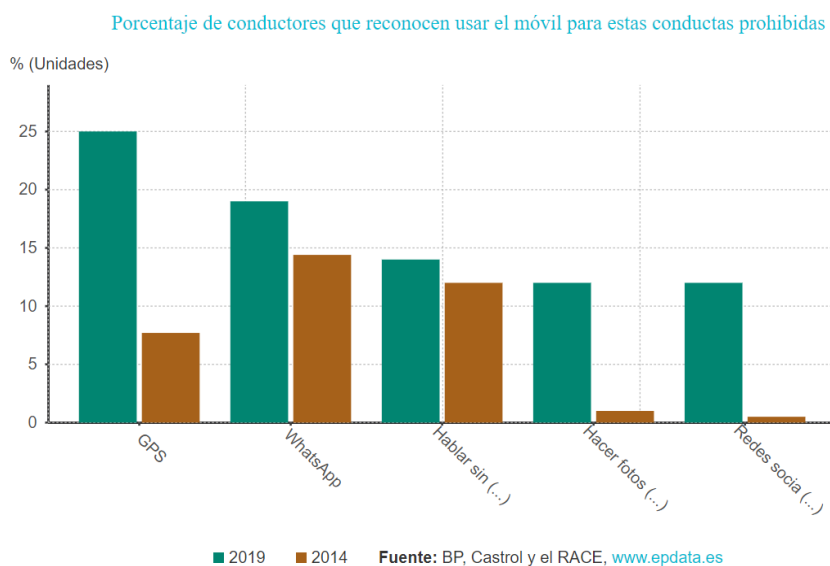


Figura 3. Porcentaje de las conductas en las que se usa el móvil al volante (EpData, 2022)

Con la vuelta a la vida cotidiana se han retomado los niveles de atascos previos a la pandemia, llegando a registrarse, el pasado día 4 de octubre de 2021 durante la hora punta de la mañana, hasta 32 kilómetros de congestión en la M-30, igualando así los datos obtenidos en fechas similares de 2019. Dicha carretera madrileña es la que más aglomeraciones y retenciones presenta, junto con la M-40 o la A-2 en sentido entrada a Madrid. [6]

Diariamente, el portal del Ayuntamiento de Madrid publica todos los días laborales, un informe que recoge lo sucedido en el día en hora punta, es decir, desde las seis de la mañana hasta las diez de la mañana de ese mismo día, en las diferentes carreteras que componen el municipio tanto urbanas como interurbanas. Dicho informe explica los accidentes, obras y vehículos averiados. Por ejemplo, el viernes 6 de mayo de 2022 se produjeron un total de 8 incidencias en hora punta, de entre las carreteras más concurridas es importante destacar la M-30 o la A-2.

[7]

Pese a que los mayores atascos se generan en las grandes carreteras, también existen atascos en el entorno urbano, principalmente en los accesos. No obstante, en la siguiente gráfica podemos ver como el tráfico en la hora punta del día 6 de mayo de 2022 fue un 42% menos que el de un viernes similar en la M-30.

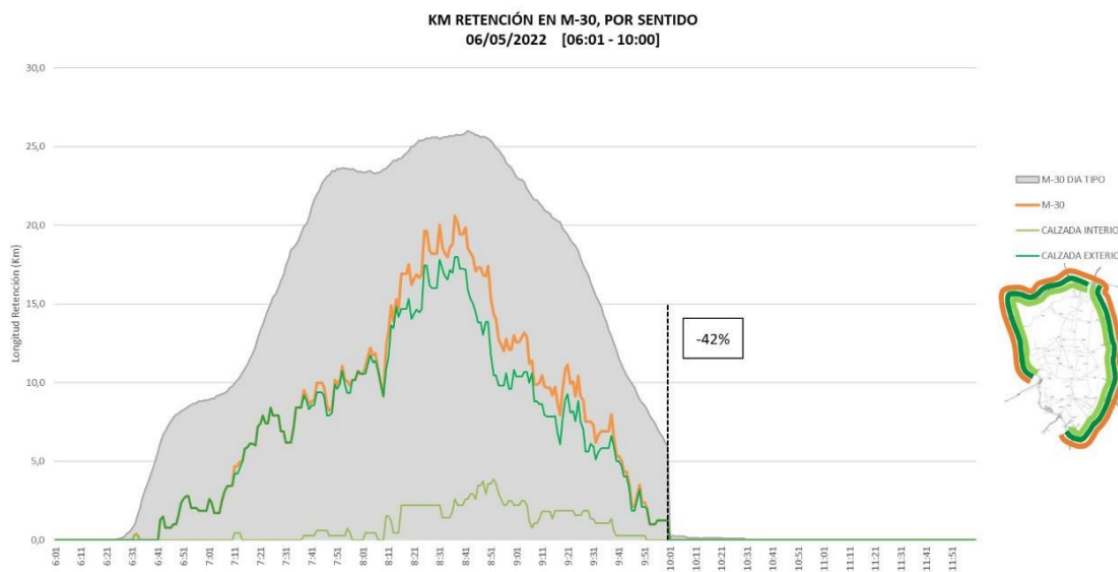


Figura 4. Evolución del tráfico en la M-30 el día 06/05/2022 (Ayuntamiento de Madrid, 2022)

Cristina López García, directora técnica del Observatorio de la Movilidad Metropolitana, asegura que tras la comodidad de circulación por las carreteras en los meses de pandemia, el uso del coche ha aumentado. Palabras que coinciden con el estudio realizado por David Lois, investigador sobre movilidad urbana en la Universidad Politécnica de Madrid. [8]

Pese a que el teletrabajo ha aumentado notoriamente desde el estallido de la pandemia. Madrid es la comunidad en la que más trabajadores usan el formato de teletrabajo de manera normalizada. Según Google, el número de personas (y por consiguiente, de vehículos) que se desplazan a sus lugares de trabajo, ha disminuido en comparación con los datos de antes de la pandemia en un 19%. No obstante, estos datos no son una anomalía, en Lisboa o Berlín hay

hasta un 20% menos llegando hasta a un 28% en Londres. Pese a la disminución del tráfico, continúa habiendo atascos en Madrid. Por lo tanto, es necesario continuar implementando y pensando nuevas medidas para intentar disminuir la ocupación en las carreteras. [8]

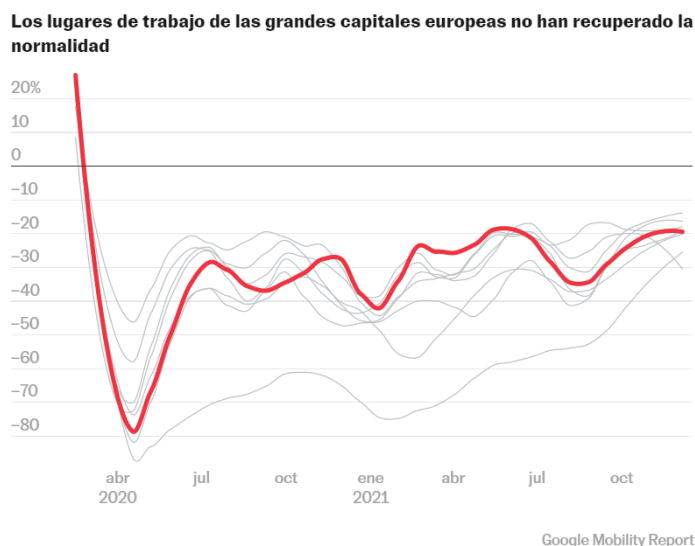


Figura 5. Evolución vuelta a la normalidad de las grandes capitales europeas (El país, 2021)

Alternativas. Transporte público

La Asociación de Transportes Públicos Urbanos y Metropolitanos o ATUC realizó una encuesta cuyos resultados dicen que el principal motivo del aumento de los viajes en coche y, por consiguiente, disminución de los usuarios que viajan en transporte público, se debe en gran medida al miedo y las precauciones ante el riesgo sanitario y los posibles contagios de covid. [8]

Una alternativa a los coches se puede encontrar en el transporte público. La Empresa Municipal de Transportes de Madrid (EMT), tiene como objetivo servir a la población de Madrid, y, para ello, debe seguir mejorando y ampliando sus servicios. La capital consta de diversas líneas de metro, autobuses y cercanías cómo medios principales para desplazarse por la comunidad. [9]

Para incitar a la gente a utilizar el transporte público, la Comunidad de Madrid ofrece ayudas como el bono transporte que, para menores de 26 años, cuesta un total de 20€ el abono joven durante un período de 30 días. Un precio muy inferior al que puede costar viajar en coche diariamente para ir al trabajo, colegio o universidad. Una noticia publicada en 2021 por el independiente comunica que la Consejería de Transporte propone extender la tarifa para jóvenes del abono transporte hasta los 30 años. [10]

Desde la finalización de la pandemia, se ha aumentado la inversión de dinero destinada a la realización de campañas para incitar el uso del transporte público haciendo énfasis en la contaminación y sostenibilidad. [11]

BiciMAD es otra de las soluciones que desde la EMT fomentan como medio de transporte por la ciudad de Madrid basado en el uso de bicicletas eléctricas. [11]

Una de las campañas más actuales que ha puesto en práctica la EMT para provocar el aumento del uso de transporte público está enfocada con la vuelta a los colegios y trabajos tras las vacaciones de semana santa. para ello la EMT anunció que sería gratis viajar por Madrid en autobús durante el lunes y martes de la semana siguiente a las vacaciones de semana santa, es decir, el 19 y 20 de abril.

Uno de los mayores criterios que los usuarios tienen en cuenta a la hora de elegir entre viajar en un vehículo propio o haciendo uso de cualquiera de las opciones existentes de transporte público en Madrid es el tiempo y la comodidad de cada una de los medios. En relación a la primera, el transporte en autobús es el único que puede variar en función del tráfico o la meteorología del día, es decir, es el único medio de transporte que se puede ver modificado en el día en función a la situación de las carreteras. Para solucionar este problema, existen diversos carriles en la capital exclusivos para autobuses o taxis que les facilitan el movimiento pese al tráfico que haya en la capital. Con respecto a la comodidad, en las horas críticas se incrementa la periodicidad de los vehículos provocando así menos congestión.

Tipos de vehículos y aumento del coste del consumo

En el mes de marzo de 2022 se realizaron 59.920 matriculaciones, un 30,18% menos con respecto al mismo mes del año anterior. En marzo, la variación anual fue al alza con un 6,56%. Datos que no son uniformes y que varían mucho en función al mes en el que nos encontremos. [12]

Variación anual en el número de turismos matriculados

	Variación anual (%) (%)
2022 - Marzo	-30,178
2022 - Febrero	6,562
2022 - Enero	0,979
2021 - Diciembre	-18,67
2021 - Noviembre	-12,296
2021 - Octubre	-20,456
2021 - Septiembre	-15,677
2021 - Agosto	-28,9
2021 - Julio	-28,855
2021 - Junio	17,101

Fuente: Anfac / Faconauto / Ganvam, www.epdata.es

Figura 6. Variación anual de matriculaciones (EpData, 2022)

De los 59.920 turismos matriculados se encuentran vehículos diésel, gasolina e híbridos, enchufables y otros como grupos principales. De los tres subgrupos, el que menos matriculaciones tiene es el de combustible diésel, seguido por los híbridos, eléctricos y otros tipos, terminando por los de gasolina. [12]

En estos últimos meses, la subida de la luz, gas, diésel y gasolina ha sido una de las noticias más comentadas en el ámbito nacional. La comparación entre el precio de la luz en 2021 y en 2022

nos ayuda a ver los disparados que se encuentran los precios actualmente y la gran subida que se produjo a finales del año 2021. [13]

Comparación diaria entre el precio de la luz en 2021 y en 2022

Datos actualizados a 7 de mayo de 2022

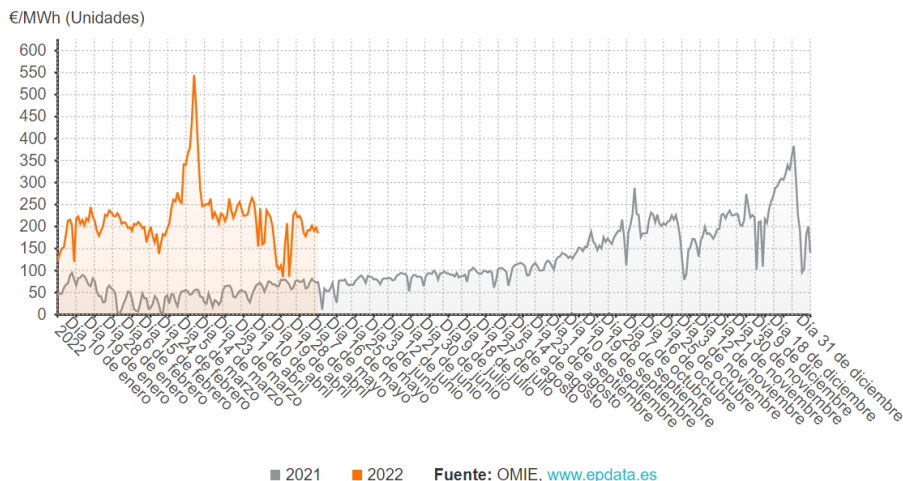


Figura 7. Evolución precio diario de la luz (EpData, 2022)

El precio de la gasolina y el gasóleo es otra cifra que ha subido en exceso estos últimos meses y que está haciendo que los viajes en coche sean más caros. En la actualidad, el precio de la gasolina es inferior al del gasóleo, dato bastante diferente a años anteriores en los que el diésel era inferior. [14]

Evolución del precio de la gasolina y el diésel en España

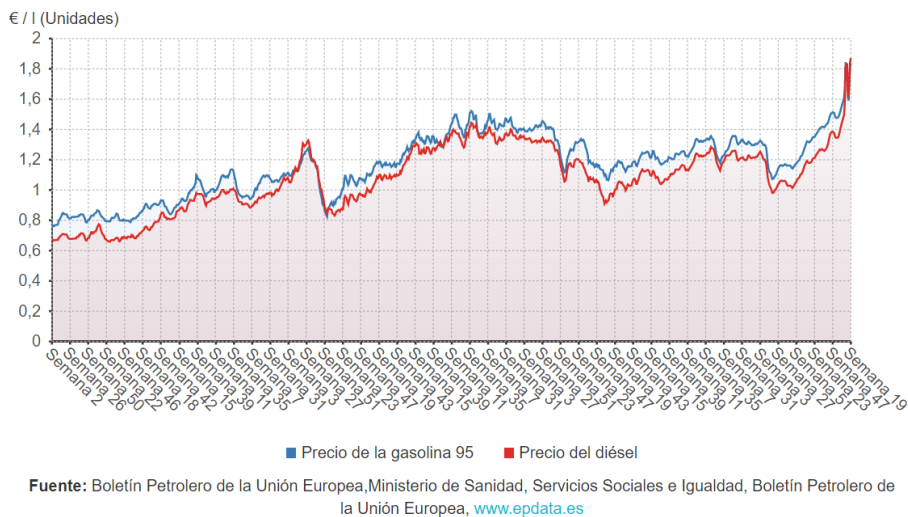


Figura 8. Evolución precio de la gasolina y diésel (EpData, 2022)

Para intentar paliar esta subida de precios en la gasolina y gasóleo, el Gobierno ha puesto en marcha un descuento de 20 céntimos por litro, de los cuales 15 céntimos los aportará el Estado y los otros cinco, en caso de ser una petrolera, los pagará la propia empresa.

Ejemplos similares

Para conocer la actualidad del tráfico existen diferentes aplicaciones. Las más significativas y conocidas por gran parte de los usuarios son las siguientes:

- **Google maps**

Dicha aplicación muestra un mapa muy completo puesto que no solo muestra el nombre de las calles sino que también aparece la localización de establecimientos o parkings. La información principal que muestra Google maps está relacionada con las diferentes rutas posibles para ir desde un origen a un destino en concreto con la estimación aproximada de tiempo de duración en el trayecto. esta información la puede dar de diferentes maneras dependiendo del modo de viaje, coche propio, transporte público, andando o bicicleta. Dependiendo del modo de transporte, la ruta varía. Además, muestra más de una opción para elegir. Como añadido a esta opción, existe la posibilidad de añadir más de un destino en el orden que se quiera hacer.

Esta información la puede dar teniendo en cuenta la ubicación actual del usuario o eligiendo también el punto de partida, es decir, se pueden visualizar rutas que se van a realizar ahora mismo o en un futuro. Una vez elegida la ruta que se quiere llevar a cabo y, si se desea hacer en ese mismo momento, esta aplicación es capaz de indicar todos los pasos a ejecutar, cuando girar en una calle, que salida coger o que hacer en una rotonda. Esta información se representa escrita en los detalles de la ruta o se puede seleccionar en “iniciar la ruta” y él mismo dependiendo de la ubicación, irá indicando qué se debe hacer.

En caso de querer ver una ruta que se vaya a usar a posteriori, se pueden seleccionar diferentes variables que ayudan a obtener un resultado más preciso al que de verdad se desea. En este apartado la aplicación tiene en cuenta que, dependiendo de la franja horaria del día, el tráfico en algunas carreteras puede variar. Para solucionar esta diferencia de tiempo, la aplicación solicita al usuario que indique a qué hora desea salir o antes de que hora quiere llegar a su destino para así, darle como resultado el tiempo total del trayecto más similar al que en realidad va a tener cuando realice el viaje señalado en la aplicación.

Otra opción en relación a las diferentes rutas que permite seleccionar la aplicación cuando da el resultado solicitado es marcar si se quiere evitar que el resultado incluya zonas de peaje o no.

Como información adicional, se puede buscar cualquier restaurante, tienda o lugar y el propio Google Maps enseña imágenes publicadas del sitio y los comentarios de usuarios, el momento en el que está más o menos transcurrido, o el horario de apertura y cierre si se trata de un establecimiento con horario.

- **Waze**

Es otra aplicación muy similar a la mencionada anteriormente, pero esta se encuentra en pleno auge estos últimos años. Entre sus diferencias con Google maps está que esta última ha sido creada por la comunidad de conductores que comparten todo tipo de información que se encuentran en su trayecto diario, es decir, avisan de accidentes y radares en mayor medida.

[15]

Sofía Corral Caballero

Entre sus características se observan como apenas existe diferencia entre las ofrecidas por Google Maps. La principal y, por la que los usuarios se descargan la aplicación es para buscar la mejor ruta entre dos ubicaciones teniendo en cuenta lo que sucede en tiempo real y teniendo en cuenta posibles accidentes u otros obstáculos como obras. A diferencia de la aplicación que ofrece Google, esta está destinada únicamente al uso en tiempo real de la aplicación.

Una vez comienza la ruta, existe un velocímetro en la parte inferior de la pantalla que ayuda a respetar los límites de velocidad y a conocer en todo momento la velocidad máxima a la que se puede circular por dicho tramo de la carretera en el que se está circulando.

Para mayor facilidad, desde la propia aplicación, se puede controlar la música o podcast que se escuchan sin tener la necesidad de salir fuera de la aplicación para cambiar de música.

Como valor añadido, se puede indicar el ir a una gasolinera, por ejemplo, y la aplicación indica la mejor ruta para viajar y, a su vez, encuentra la más cercana teniendo en cuenta la ubicación en la que se encuentra el vehículo en ese momento. Algo similar sucede en el caso de querer localizar un aparcamiento cercano al lugar de destino, la propia aplicación va a ayudar a encontrarlo.

Por último, está integrado un calendario en el cual se puede indicar la hora a la que ir a un lugar en concreto, y la aplicación avisa cuando se debe salir de casa para llegar a tiempo mediante un aviso en el teléfono móvil a modo de notificación de la aplicación.

Como la aplicación funciona en base a la información que los usuarios que la utilizan aportan, es decir, si no hay ningún usuario la aplicación no daría información real acerca de la trayectoria, desde Waze quieren que sus clientes utilicen siempre la aplicación para viajar independientemente de si se conocen o no el trayecto con la finalidad de obtener más datos. Para ello, han creado una nueva versión en la que tras recorrer 160 kilómetros podrás desbloquear una nueva expresión de sentimientos.

En definitiva, aparte de la manera de obtener la información pretenden diferenciarse de Google Maps intentando personalizar la experiencia de conducción de sus clientes en las diferentes rutas. [16]

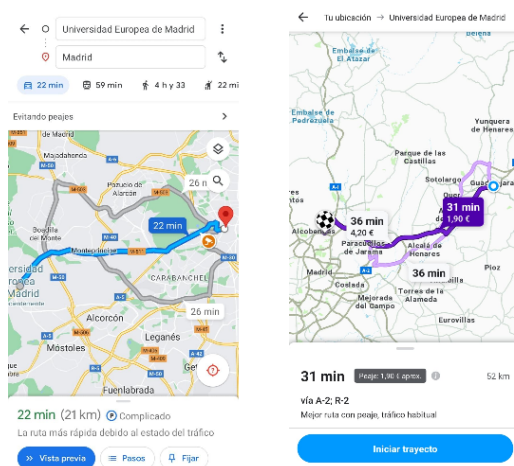


Figura 9. Ejemplo aplicación Google maps y Waze (2022)

2.2 Contexto y justificación

¿Existen las suficientes medidas para disminuir los atascos en Madrid? En la actualidad se puede observar como pese a todas las campañas realizadas para incrementar el uso de transporte público, los ciudadanos continúan utilizando en gran medida el transporte en vehículo propio provocando grandes colapsos, sobre todo en las horas puntas de entrada y salida al trabajo.

Con la creación de esta aplicación se pretende ayudar a evitar los atascos y, por consiguiente, se ahorraría energía. Para ello vamos a intentar predecir al cliente de la aplicación la hora idónea para viajar a ese punto con la finalidad de distribuir el tráfico que circula por la zona.

La probabilidad de que se produzca un atasco crece cuanto mayor sea el número de coches que circulan por la carretera en ese momento y, como efecto secundario, los atascos ocasionan un mayor atasco. Con esta aplicación, se pretende distribuir el número de coches que circulan por un punto en concreto en un día, haciendo que disminuya la probabilidad de colapso y atasco.

Los usuarios usarán esta aplicación en su vida cotidiana, pues podrán observar en ella tanto datos en tiempo real del estado de las carreteras de una manera visual fácil de entender para todo el público, como pedir consejo para circular un día en concreto por una carretera.

Alguno de los casos de uso que esta aplicación puede solucionar pueden ser:

- **Mejor hora para pedir una cita:** Necesito pedir una cita y quiero evitar atascos en el trayecto, puedo meterme en la aplicación y ver las horas menos concurridas de ese día teniendo en cuenta el día de la semana al que pertenece.
- **Quiero ir a este sitio:** Me gustaría ir a visitar esta tienda o a ver algún monumento este día de la semana que tengo libre, pero quiero saber la hora en la que menos atasco haya para así ahorrar energía, tiempo, y, por supuesto, tener menos posibilidad de colisión en el trayecto.
- **Conocer la carretera con más ocupación en tiempo real:** En este caso, la información que se le ofrece al cliente de la aplicación es en tiempo real y está enfocada a ver las zonas de Madrid cuya intensidad de coches en la carretera es elevada, ayudando a tener más precaución si va a pasar por un tramo con mucha ocupación pues deberá reducir la velocidad y esto en ocasiones es motivo de accidente. Además, es información que nuestro cliente podrá ver antes de comenzar su trayecto, sin necesidad de hacer uso del dispositivo móvil mientras está al volante.
- **Análisis de patrones:** Con la aplicación no solo se puede ver la mejor hora para viajar a un lugar, sino que se ha implementado un apartado con diferentes gráficas que ayudan a detectar más conclusiones y análisis de los datos. En ellos se puede ver si existe una relación entre diferentes semanas o qué día de la semana es en el que menos tráfico circula, entre muchas otras conclusiones.
- **Quiero cambiar de trabajo:** Me han ofrecido una nueva oferta laboral y me gustaría saber si me conviene aceptarla o no. Entre uno de los factores que me van a ayudar a tomar la decisión está el trayecto diario a la oficina. La aplicación es capaz de enseñar la hora menos concurrida y mediante una serie de análisis y gráficos determinar si tendrá atasco diariamente o no, y consecuentemente, en qué medida.

- **Quiero cambiar de lugar de vivienda:** Este caso es similar al anterior, pero no solo se deberá analizar las horas puntas de entrada y salida del trabajo sino el cómputo entero del día. Para ello, nos ayudaremos de una serie de gráficas que mostraremos en la aplicación haciendo referencia al análisis de los datos obtenidos del tráfico del municipio de Madrid.

Con el uso de la aplicación, a parte de buscar llegar lo antes posible a un sitio evitando atascos, se pretende disminuir el consumo de energía y distribuir los coches en todas las horas del día sin tener la necesidad de congregarse en la hora punta de la salida o entrada a los trabajos.

Como valor añadido o diferenciador del resto de aplicaciones, se debe tener en cuenta que un alto porcentaje de personas usan el dispositivo móvil al volante para el GPS, este acto está prohibido y sancionado con hasta puntos del carnet. Con la aplicación, no será necesario usarla mientras estamos viajando ya que la mayoría de las utilidades que nos presenta las podemos obtener antes de comenzar el trayecto y así no vernos en la necesidad de incumplir las normas impuestas por la DGT.

A modo de conclusión, las medidas tomadas hasta ahora para evitar atascos e intentar reducir el consumo de energía no son suficientes. Por ello, parte la idea de crear una aplicación que detecte zonas de atasco siguiendo un patrón de datos y pueda informar a sus clientes antes de comenzar el trayecto sin necesidad de usar el móvil al volante.

2.3 Planteamiento del problema

Una vez analizado el tráfico en el municipio de Madrid y, tras llegar a la conclusión de que no existen medidas suficientes para evitar las grandes aglomeraciones de coches que se producen a hora punta, existe la idea de crear una aplicación que, de una manera diferente a las ya existentes, se intente disminuir este tráfico.

Con la reducción de las aglomeraciones en Madrid no solo se pretende eliminar el número de accidentes, si no que también se quiere evitar que todo aquel usuario que no tenga la necesidad de viajar en las horas más concurrentes del día, lo haga. Además, se pretende dar toda la información necesaria antes de comenzar el trayecto, para así no incumplir las normas de tráfico, que no permiten manipular el dispositivo móvil cuando se está al volante.

Para ofrecer al usuario información suficiente como para poder resolver todo lo que necesite, se ofrecen diversas vistas. Una de ellas, con los datos en tiempo real en los que se puede determinar aquellas zonas del municipio con más ocupación para así ayudar al cliente de la aplicación en la toma de decisiones para viajes inmediatos.

También, se ofrece un análisis más extenso del tráfico en los diferentes puntos de Madrid en los que hay un sensor que mida la intensidad de este, con la finalidad de poder detectar patrones durante las semanas y tomar decisiones más enfocadas en el largo plazo y no en un momento concreto. Por último, se pretende ayudar al usuario a disminuir su consumo de energía con la creación de un modelo capaz de indicar la mejor hora para viajar al destino deseado en el día marcado.

Con toda esta información que se pretende dar al usuario se quiere distribuir el flujo de coches a lo largo del día, repartiendo el tráfico diario para conseguir un equilibrio y así disminuir la ocupación de las carreteras en la hora punta.

Capítulo 3. OBJETIVOS

3.1 Objetivos generales

El objetivo principal es generar una aplicación haciendo uso de los datos de tráfico recopilados en tiempo real sobre la ciudad de Madrid, empleando la metodología scrum para su desarrollo y cuya finalidad principal es hacer que los clientes ahorren energía y tiempo de desplazamiento con el vehículo.

3.2 Objetivos específicos

- Determinar la fuente más fiable para la recopilación de datos en tiempo real. Para ello, se evalúan las diferentes fuentes analizadas teniendo en cuenta la calidad y cantidad de datos que aportan, su fuente de obtención y la complejidad para extraerlos.
- Recopilar la información suficiente de los datos de tráfico del municipio de Madrid en tiempo real haciendo uso de herramientas propias del Big Data como es la nube, en concreto, Google Cloud Platform.
- Tratar, limpiar y realizar un análisis exploratorio de los datos recopilados.
- Ser capaz de obtener conclusiones y verificar la veracidad de los datos mediante el uso de herramientas de visualización.
- Implementar un modelo capaz de predecir la hora idónea a la que pasar por un punto concreto con precisión en sus resultados.
- Crear una aplicación que represente todo el trabajo realizado e implementarla como producto final para los usuarios.
- Generar una base de datos para almacenar los datos obtenidos en el registro por los usuarios de la aplicación.
- Emplear metodologías ágiles para el desarrollo del trabajo al completo.
- Aplicar todos los conceptos adquiridos a lo largo de la carrera en el desarrollo del proyecto.

3.3 Beneficios del proyecto

Teniendo en cuenta, principalmente, el tráfico en la capital, lo valioso que es el tiempo personal, los accidentes y la subida de la luz y el carburante, con la creación de esta aplicación, los usuarios serán capaces de, en primer lugar, disminuir el uso de la energía de desplazamiento de su vehículo y, secundariamente pero no menos importante, al haber disminuido los atascos en Madrid, reducir el número de accidentes o la probabilidad de que estos sucedan será inferior.

También se pretende atenuar el uso del móvil al volante, siendo capaces de informar a nuestro usuario del tráfico en tiempo real, así como a largo plazo antes de comenzar el trayecto, ayudando a rebajar la tasa de accidentes.

Además, la aplicación creada es multiplataforma, fácil de entender para los usuarios, con una respuesta rápida a las diferentes consultas o resultados y de diseño responsive.

Capítulo 4. DESARROLLO DEL PROYECTO

4.1 Planificación del proyecto

Para llevar a cabo el proyecto, se ha seguido una metodología scrum con un total de siete sprints, cuya duración es de tres semanas cada uno. A lo largo del proyecto se han realizado las siguientes actividades divididas en diferentes bloques:

Bloque 1: Actividades a realizar antes de comenzar con el desarrollo del proyecto

- Contextualizar el proyecto para comprender el problema que se quiere resolver y ver las posibles soluciones que se pueden aplicar. Para ello, es muy importante tener en cuenta los proyectos similares que ya existen e intentar buscar una mayor mejora o una diferenciación. Además, se ha realizado un análisis previo del proyecto identificando todas las herramientas que se iban a poner en uso a lo largo del proyecto, la dificultad de cada una de las tareas y una estimación de tiempo
- Realizar un anteproyecto donde se tenga en cuenta los antecedentes, problemas que se quieren resolver, contextualizar la idea que se va a llevar a cabo, la metodología a utilizar y definir el plan del proyecto a cumplir durante su ejecución.
- Ampliar el anteproyecto con conceptos no solicitados para la entrega, pero necesarios de cara al resultado final como la definición del alcance o los requisitos tanto funcionales como no funcionales.
- Preparar el entorno: una vez que conocemos los requisitos, objetivos y alcance, se procede a preparar las herramientas a utilizar en el desarrollo del TFG. Además, al utilizar conceptos de herramientas no usadas con anterioridad, fue necesario una formación haciendo uso de diversos vídeos, cursos y artículos.

Bloque 2: Desarrollo del proyecto

1. Obtención y análisis de los datos en local

- Determinar el conjunto de datos idóneo para nuestro objetivo. Para ello, se debe estudiar las diferentes opciones que existen y elegir la correcta en función principalmente de la facilidad para obtener los datos y la información que nos aportan las variables que conforman el conjunto.
- Implementar proceso de scraping de los datos para un momento concreto, es decir, de forma genérica para luego planificarlo de forma cíclica y recopilar los datos en ventanas temporales. Con esta actividad se genera la función final de scraping, recopila los datos a obtener en un futuro en tiempo real y se analizan mientras se trabaja en otras actividades en paralelo.
- Monitorizar el proceso para recoger los datos en tiempo real. Una vez obtenida la función general, se debe crear un proceso que haga que la función se ejecute de forma periódica y así obtener los datos en tiempo real.
- Analizar los resultados obtenidos para verificar que se han recogido y almacenado correctamente y que los datos tienen sentido.

2. Obtención y análisis de los datos haciendo uso de herramientas en la nube (GCP)

- Implementar la función creada y probada en local, capaz de recoger el conjunto de datos de tráfico. Seguidamente, utilizar Google Cloud Platform (GCP) para realizar esta tarea de forma planificada, haciendo uso de una de sus herramientas llamada Cloud Function
- Crear un espacio dónde almacenar los datos obtenidos en la propia plataforma de GCP de cara a tener un histórico de información usando Cloud Storage y creando un Bucket dentro.
- Planificar el proceso mediante la herramienta Google Scheduler y ejecutar la función cada 10 minutos al día para recoger los datos.
- Unificar y relacionar el proceso de petición de llamada generado en Google Scheduler con la función creada para este proyecto en Cloud Function y, posteriormente, mandar los datos recopilados a una dirección dentro de la nube que almacena toda la información recopilada en un contenedor.
- Analizar los resultados obtenidos y corroborar que son fiables.

3. Crear el modelo

- En primer lugar, se ha creado el formulario que el usuario debe rellenar para la predicción. En dicho formulario se indica al usuario que introduzca la calle a la que quiere ir, el día de la semana que lo va a hacer y la franja horaria a la que le gustaría ir. El modelo le devolverá la mejor hora para pasar por ese punto evitando atascos.
- En función a los valores rellenados en el formulario, se filtra de todo el conjunto de datos aquellos que cumplan con los criterios rellenados en el formulario.
- Se genera el modelo en función a los datos filtrados mediante el uso de cálculos matemáticos.

Bloque 3: Visualización

- Haciendo uso de la herramienta Power BI, se ha desarrollado un dashboard que ayuda a visualizar el conjunto de datos obtenidos, mostrando algunos KPIs que se determinan importantes, así como diferentes visualizaciones, de cara a verificar la veracidad de la información proporcionada por los datos en el scraping en tiempo real.
- Utilizando el framework ionic se desarrollará un mapa en la aplicación con los datos en tiempo real.
- Usando de nuevo Power BI, se implementarán diferentes gráficos que formarán parte de la segunda página de la aplicación mencionada anteriormente.

Bloque 4: Crear la aplicación o producto para el usuario del proyecto

- Generar un prototipo que recoja todas las ventanas que va a tener la aplicación final y el contenido de cada una de ellas. Además, se definirá el estilo de las diferentes ventanas y su interacción dentro de la aplicación.
- Crear la base de la aplicación, es decir, montar las diferentes vistas que va a tener la aplicación y su navegación.
- Generar la parte de registro y login de la aplicación

Sofía Corral Caballero

- Conectar las vistas de registro y login a una base de datos que almacene la información de los usuarios una vez se registren y se verifique que el usuario y contraseña sean correctos al hacer el login.
- Conectar los datos almacenados en GCP con la aplicación y realizar pequeñas modificaciones que ayuden a representarlos.
- Implementar un mapa en la primera vista de la aplicación representando en diferentes colores la ocupación de las carreteras en tiempo real.
- Añadir diferentes gráficos creados en Power BI en la segunda vista del producto para ayudar a los usuarios a obtener información sobre el tráfico en Madrid en tiempo real.
- Implementar el modelo que ayude al usuario a determinar la mejor hora para pasar con el coche por un destino concreto teniendo en cuenta que nuestro objetivo principal es ahorrar energía y tiempo con el vehículo. Para ello, se implementará también un formulario.
- Finalmente, se desplegará la aplicación para que todos los usuarios puedan disponer de ella.

Bloque 5: Pruebas

- Se realizarán pruebas de los resultados obtenidos en el modelo para comprobar su comportamiento
- Se realizarán pruebas de la aplicación para comprobar que funciona para Android e ios, que la interacción entre las vistas es correcta o que los datos son correctos, entre muchas más comprobaciones.

Antes de representar esquemáticamente la planificación por semanas llevada a cabo definitivamente, es importante destacar la realización de actividades en paralelo y la implementación de la documentación en cada uno de los apartados. Además, el número total de horas personales invertidas en el proyecto ha sido de 450h. La planificación que se ha llevado a cabo ha sido la siguiente:

	Enero	Febrero	Marzo	Abril	Mayo	Junio
ANÁLISIS PREVIO						
Redactar idea del proyecto						
Contexto de la idea						
Estado del arte y antecedentes						

Sofía Corral Caballero

PLANIFICACIÓN												
Definir el alcance del proyecto												
Definir objetivo principal												
Definir objetivos secundarios												
Definir requisitos funcionales												
Definir requisitos no funcionales												
Planificar la arquitectura												
PREPARAR ENTORNO												
Instalar y comprobar el funcionamiento												
Formación en ciertas herramientas												
DESARROLLO DEL PROYECTO												
<i>Obtención y análisis de los datos en local</i>												

Buscar la/las fuentes del conjunto de datos											
Crear proceso de obtención del conjunto de datos para un momento en concreto											
Monitorizar el proceso de obtención de datos para recogerlos en tiempo real											
Analizar los resultados obtenidos											
<i>Obtención y análisis de los datos haciendo uso de herramientas en la nube (GCP)</i>											
Implementar la función creada en local en Cloud Function											
Crear un Bucket en Cloud Storage											
Implementar un temporizador											

Sofía Corral Caballero

con Google Scheduler																				
Unificar los procesos creados en las diferentes herramientas																				
Analizar los resultados obtenidos																				
Crear el modelo																				
Crear formulario																				
Preparar los datos																				
Crear modelo																				
VISUALIZACIÓN																				
Crear diferentes visualizaciones para el análisis de los datos																				
Crear un mapa para la APP																				
CREAR APLICACIÓN																				
Generar prototipo																				

PASOS FINALES												
Entrega primera versión de la memoria al tutor												
Realización de los cambios sugeridos por el tutor en la memoria												
Entrega versión final de la memoria												
DEFENSA DEL PROYECTO												
Defender el proyecto												

Tabla 1. Planificación del proyecto por actividades y semanas de cada mes

4.2 Descripción de la solución, metodologías y herramientas empleadas

4.2.1 Explicación de la solución

El proyecto está dividido en diferentes bloques. El primero de ellos está relacionado con los datos. En él se engloba todo el proceso de ingesta de datos, preprocesado de los datos, análisis de los mismos e implementación de diferentes visualizaciones con la finalidad de poder obtener diferentes conclusiones y estudiar el comportamiento a futuro.

En el segundo bloque se ha aplicado un modelo para predecir el momento del día idóneo teniendo en cuenta las variables rellenas por el usuario de ubicación, día de la semana y franja horaria. Dentro de este apartado se encuentra todo el proceso de creación de un modelo de aprendizaje automático, su entrenamiento y evaluación de los resultados.

Finalmente, el último bloque consta del desarrollo de una aplicación dónde el usuario sea capaz de ver fácilmente, por un lado, el tráfico en tiempo real y por otro, pueda hacer un análisis del histórico de esos datos e incluso predecir la hora más idónea a la que circular, es decir, se implementará todo el trabajo previo realizado.

Dentro de los diferentes bloques, es importante destacar las siguientes actividades:

Conjunto de datos:

Durante toda la realización del proyecto, se ha partido de un único conjunto de datos obtenido gracias a la realización de diferentes técnicas de obtención de datos sobre una única fuente en concreto.

La fuente sobre la que se parte pertenece al portal de datos abiertos del ayuntamiento de Madrid. En la propia página existen diferentes catálogos de datos, se ha seleccionado aquel que habla sobre el tráfico y aporta datos en tiempo real. [17]

En concreto, el conjunto de datos seleccionado aporta información gracias a la sensorización del tráfico en la comunidad, al conteo de vehículos y a la obtención del grado de ocupación de cada una de las vías en las que está localizado un sensor capaz de medir dicha información.

La información en la fuente se actualiza cada cinco minutos aproximadamente, ya que se considera como tiempo necesario para proporcionar unos valores reales sin verse afectados por los semáforos, al no ser valores precisos, y para evitar duplicidad de información, se ha decidido que en el proyecto, se recuperarán los datos cada diez minutos, llegando así a tener un total de 144 ficheros por día.

Existen aproximadamente 7.800 detectores de vehículos, divididos la mayoría en sistemas básicos de control de semáforos y una parte minoritaria compuesta por sistemas óptimos de visión artificial, sensores específicos para vías rápidas y acceso a la ciudad. Gracias a todos estos detectores de vehículos, existen en Madrid alrededor de 4.100 puntos de medida repartidos en diferentes posiciones de Madrid, tanto urbanos como interurbanos.

Las variables que componen nuestro conjunto de datos se han obtenido mediante un fichero XML con una cabecera principal y dos subapartados que diferencian los datos entre urbanos e interurbanos.

La cabecera nos da la información de la fecha y hora en la que se obtuvieron los datos en una única variable. Esta información es genérica, es decir, engloba los datos urbanos y los interurbanos.

Campo	Descripción	Ejemplo
fecha_hora	fecha y hora de obtención de los datos	07-04-2022 14:43:04

Tabla 2. Nombre del campo, descripción y ejemplo de las variables genéricas para los datos urbanos y no urbanos

Para el conjunto de datos de tráfico obtenidos mediante el XML en tiempo real de cada punto de medida situado únicamente en una zona urbana, podemos obtener la información del tráfico mediante las siguientes variables:

Campo	Descripción	Tipo
idelem	Es el identificador del punto de medida	numérico

descripcion	Nombre del punto de medida. En este apartado se tiene en cuenta el nombre de la calle (en caso de que esta fuera muy larga y exista más de un sensor en la misma dirección, se especificará el intervalo de la calle que engloba la información), el número de la calle en el que se encuentra en concreto el sensor situado y la dirección de la carretera (indicándolo en este caso mediante los puntos cardinales y entre paréntesis con el nombre de las dos calles que conectan en sus extremos en orden de dirección)	texto
accesoAsociado	Código de control relacionado con el control semafórico para la modificación de los tiempos	numérico
intensidad	Intensidad del número de vehículos que circulan por hora. Un número negativo implica la ausencia de datos	numérico
ocupacion	Porcentaje de ocupación del punto de control por vehículos. Un número negativo implica la ausencia de datos	numérico
carga	Es un parámetro de carga del vial que se calcula en función de la intensidad, ocupación y características de la infraestructura. Un número negativo implica la ausencia de datos	numérico
nivelServicio	Nivel del servicio	numérico

intensidadSat	Intensidad de la saturación de la vía reflejada en vehículos por hora. Este dato nos indica el número máximo de vehículos que pueden pasar en el acceso a la intersección teniendo en cuenta que se mantiene la fase verde del semáforo	numérico
error	Código que controla la validez de los datos del punto de medida	texto
subarea	Identificador de la subárea de explotación de tráfico a la que pertenece el punto de medida	numérico
st_x	Coordenada X medida en UTM	numérico
st_y	Coordenada Y medida en UTM	numérico

Tabla 3. Nombre del campo, descripción y tipo de las variables específicas para los datos urbanos

Un ejemplo que representa la estructura de la información recogida en uno de los puntos de medida es:

```
<pm>
  <idelem>3409</idelem>
  <descripcion>SEPULVEDA Ø118 N-S (CEBREROS-CJAL. FCO. J.
  JIMENEZ)</descripcion>
  <accesoAsociado>240102</accesoAsociado>
  <intensidad>424</intensidad>
  <ocupacion>3</ocupacion>
  <carga>17</carga>
  <nivelServicio>0</nivelServicio>
  <intensidadSat>3000</intensidadSat>
  <error>N</error>
  <subarea>1718</subarea>
  <st_x>436004,103195086</st_x>
```


<st_y>4472588, 47356732</st_y>

</pm>

Para el conjunto de datos de tráfico obtenido mediante el XML en tiempo real de cada punto de medida situado únicamente en una zona interurbana, como pueden ser vías rápidas, accesos a Madrid o la propia M30 principalmente, se obtiene la información del tráfico mediante las siguientes variables:

Campo	Descripción	Tipo
idelem	Es el identificador del punto de medida	numérico
intensidad	Intensidad del número de vehículos que circulan por hora. Un número negativo implica la ausencia de datos	numérico
ocupacion	Porcentaje de ocupación del punto de control por vehículos. Un número negativo implica la ausencia de datos	numérico
carga	Es un parámetro de carga del vial que se calcula en función de la intensidad, ocupación y características de la infraestructura. Un número negativo implica la ausencia de datos	numérico
nivelServicio	Nivel del servicio	numérico
velocidad	Velocidad media de los vehículos detectados en el tiempo de integración medida en kilómetros por hora	numérico
error	Código que controla la validez de los datos del punto de medida	texto
st_x	Coordenada X medida en UTM	numérico

Sofía Corral Caballero

st_y	Coordenada Y medida en UTM	numérico
------	----------------------------	----------

Tabla 4. Nombre del campo, descripción y tipo de las variables específicas para los datos no urbanos

Un ejemplo de dicha información representada en el XML con el que vamos a recopilar los datos es el siguiente:

```
<pm>
  <idelem>7014</idelem>
  <intensidad>420</intensidad>
  <ocupacion>4</ocupacion>
  <carga>26</carga>
  <nivelServicio>0</nivelServicio>
  <velocidad>77</velocidad>
  <error>N</error>
  <st_x>443015,212979909</st_x>
  <st_y>4481802,10519696</st_y>
</pm>
```

Extracción de datos:

El proceso de extracción de datos se ha compuesto de diferentes pasos. Siguiendo la metodología ágil scrum implementada durante todo el proyecto, los pasos vendrán identificados en un primer momento por la obtención de un conjunto de datos con el que poder comenzar a implementar el resto de tareas, modelo o visualización, entre otras. El resto de los pasos están enfocados a mejorar el conjunto de datos, añadir variables necesarias para el desarrollo, modificar pequeños errores en el proceso de scraping o implementar un script que recoja los datos en tiempo real.

Dentro del proceso de obtención de datos se encuentran dos grandes apartados. En primer lugar, se han realizado todas las codificaciones y la obtención de ficheros de ejemplo en local mediante el uso de Jupyter Notebook, pero al tener que captar la información en tiempo real y querer almacenarla, nos encontramos con la necesidad de hacer uso de herramientas específicas para Big Data, es decir, la nube. En concreto, hemos usado Google Cloud Platform (GCP).

Por lo tanto, antes de pasar directamente a la nube, se realizaron diferentes pruebas en local creando una función que, al ejecutarse, se descarga un fichero en un directorio concreto con los datos proporcionados por el ayuntamiento de Madrid y recopilados en ese mismo instante. Como se ha comentado anteriormente, la información es proporcionada en formato XML. Para obtener dicha información del enlace del Ayuntamiento de Madrid a nuestro equipo, hemos utilizado la librería *requests* de Python, que trabaja con peticiones HTTP para extraer los datos. Posteriormente, para organizar la información, se ha usado *Beautifulsoup*, una librería de

Python que analiza documentos en formato HTML, XML, etc. y los parsea, creando un árbol con todos los elementos del documento. La finalidad principal de dicha librería es extraer información de sitios web, es decir, hacer web scraping.

Una vez que se dispone de la información ordenada, es necesario estudiar la estructura que esta tiene y, se aprecia que, exceptuando la variable fecha que forma parte del encabezado del XML y que únicamente aparece una vez por cada petición, el resto de variables siguen el mismo formato independientemente de si el dato es urbano o no, a excepción de las variables de cada tipo.

Cada sensor viene separado por el comienzo de una nueva etiqueta (<pm>) y en su interior se encuentran diferentes etiquetas con el nombre de cada variable seguido del dato y, finalmente la etiqueta de cierre de cada una de las variables. Una vez dada la información de todas las variables aparece de nuevo la etiqueta de cierre (</pm>). Este proceso se repite para cada uno de los puntos de medida existentes.

Tras conocer la estructura de nuestro XML, procedemos a crear un bucle que vaya almacenando el valor de cada una de las variables en un data frame. Las diferentes dificultades o modificaciones que se han encontrado y hecho al realizar este bucle has sido las siguientes:

- Pese a que el conjunto de datos está compuesto por registros que pertenecen a puntos de medida urbanos y otros que pertenecen a los interurbanos y, a su vez cada tipo tiene un número diferente de variables, se ha decidido crear el data frame con todas las variables y dejar sin rellenar aquellas que no pertenezcan a dicho grupo. Por ejemplo, la variable velocidad únicamente se recoge cuando el dato es interurbano por lo que en el data frame aparecerá sin rellenar cuando el dato es urbano y aparecerá su valor correspondiente cuando el dato es interurbano y así con el resto de variables que únicamente pertenecen a un subgrupo. Este apartado es muy importante de cara a las visualizaciones posteriores y al análisis de los mismos, sobre todo cuando realizamos cálculos matemáticos haciendo uso del conjunto entero de datos. Siguiendo con el ejemplo de la velocidad, dato que únicamente se recoge en caso de que el sensor esté situado en una zona no urbana y rellenamos con un valor '0' cuando el sensor sea urbano, si se quiere realizar la media de esta, únicamente se deben coger los valores interurbanos para dar un resultado correcto de la información adquirida.
- Para diferenciar mejor el tipo de dato que se está recopilando en cada una de las filas (urbano e interurbano), se ha añadido una nueva variable binaria bajo el nombre *urbano* que me devuelva 'Y' en caso de ser urbana y 'N' en caso de no serlo.
- En cuanto a la variable fecha, para optimizar la función y no coger todo el fichero XML que nos proporcionaban, se decidió que, como los datos se iban a recoger en tiempo real, la fecha y hora iba a ser similar a la de ese momento por lo que procedimos a utilizar un módulo que nos proporciona python denominado *datetime* que nos devuelve la fecha y hora de ese mismo instante y así incluirlo a la variable creada en el data frame para ese apartado.
- La mayor complicación que ha surgido al recoger los datos, esto se debe a que, al indicar a nuestra función que seleccione todo el dato que se encuentre entre sus etiquetas, es decir, si cada campo de cada variable venía dado de la siguiente forma:

`<idelem>7014</idelem>`

Este devuelve el valor entre corchetes y comillas simples, es decir, siguiendo con el mismo ejemplo, el valor sería:

`['7014']`

Por lo tanto, se obtenían todos los valores del data frame entre corchetes y comillas. Para resolver este problema se tuvo en cuenta que, tras hacer el `find_all` para sacar el texto, el siguiente paso era hacer el `append` y, aquí es donde se debe indicar que únicamente se quiere introducir el valor que se encuentra en la posición '0'. El resultado que obtenido finalmente es un data frame similar al que se muestra en la siguiente figura, con un total de 4.100 filas aproximadamente y 15 columnas que engloban tanto las variables urbanas como las interurbanas:

Fecha_hora	Urbano	accesoAsociado	carga	descripcion	error	idelem	intensidad	intensidadSat	nivelServicio	ocupacion	st_x
30-03-2022 19:15:22	Y	240102	11	SEPULVEDA 0118 N-S (CEBREROS-CJAL. FCO. J. JIM...	S	3409	174	3000	0	5	436004,103195086
30-03-2022 19:15:22	Y	240101	14	CJAL. FCO. J. JIMENEZ 0126 E-O (BERLANAS-SEPUL...	S	4739	294	3000	0	5	436039,395885266
30-03-2022 19:15:22	Y	240604	15	CJAL. FCO. J. JIMENEZ 086 O-E (F. CALVO-ALHAMBRA)	S	4740	292	3000	0	6	436728,94576426
30-03-2022 19:15:22	Y	240602	9	CJAL. FCO. J. JIMENEZ 076 E-O (HURTUMPASCUAL-A...	S	4741	184	3000	0	3	436894,776506751
30-03-2022 19:15:22	Y	241101	16	ALHAMBRA 023 N-S (CJAL. FCO. J. JIMENEZ -CUART...	S	4742	323	3000	0	6	436809,990969673

Figura 10. Ejemplo data frame datos extraídos.

Una vez que se tiene los datos ordenados en un data frame, se debe indicar que guarde esta información en una dirección. Al estar trabajando todavía en local, se ha decidido almacenarlo en una carpeta del propio equipo. Para ello, hemos usado un método que nos proporciona la librería `pandas` en python (`to_excel`) que guarda el fichero en formato `.xlsx` bajo el nombre que se le indique y en la dirección también señalada por el usuario.

En cuanto a la dirección, se debe tener en cuenta que los datos se almacenan particionados y que el formato que tiene está compuesto de una carpeta por día y, dentro de cada una de las carpetas se almacena un fichero cada diez minutos haciendo un total de 144 ficheros totales por día que recogen las variables proporcionadas por el ayuntamiento de Madrid sobre el tráfico en cada uno de los puntos de medida que hay colocados por la localidad. El esquema general que siguen es el siguiente:



Figura 11. Esquema del almacenamiento de los ficheros

Para que el proceso de web scraping no fuera manual y, antes de pasar a utilizar una mejor herramienta enfocada en el tratamiento y almacenamiento de grandes cantidades de datos, se procede a realizar diferentes pruebas en local.

Para ejecutar el mismo código en python cada 10 minutos utilizamos el método `time.sleep` que lo que hace es pausar la ejecución durante el número de segundos que se le haya indicado en el argumento del método. Por lo tanto, creamos un bucle que ejecute constantemente la función y para el tiempo asignado en el método.

Realizar el proceso en local presenta varios inconvenientes que han hecho rectificar la idea principal de obtención de datos, para cambiar de herramienta a una en la nube, enfocada en Big Data. Algunas de las desventajas que encontradas son:

- Al ejecutarse en local, para que continúe funcionando Jupyter, es necesario tener el ordenador encendido durante todo el tiempo de la ejecución y, en este caso, al estar cogiendo la información en tiempo real el equipo debería estar encendido constantemente. Esto es algo imposible y poco útil a largo plazo.
- Al almacenar los datos en el equipo en local cada 10 minutos, con un total de 144 ficheros por día, es necesario contar con un espacio de memoria suficiente para el crecimiento que surgirá con el paso de los días. Estamos hablando aproximadamente de 1 GB de datos por semana que, a corto plazo puede ser viable, pero de nuevo si queremos enfocar el proyecto a futuro no es ni recomendable ni útil.

Por consiguiente y, pese a tener todo preparado, se ha decidido cambiar de herramienta. No obstante, es importante tener en cuenta que no ha sido necesario empezar de cero de nuevo ya que gran parte del trabajo realizado en local nos ha servido para utilizarlo de nuevo en Google Cloud Platform.

Para configurar el proceso basado en la extracción de datos en tiempo real (cada 10 minutos) y el almacenamiento de los ficheros en la propia nube, ha sido necesario hacer uso de diferentes servicios que el propio Google Cloud Platform proporciona. Estos son:

- **Cloud Storage:** En este servicio se ha creado un bucket para almacenar diariamente los datos. Dentro del contenedor hay una carpeta por día y en su interior los ficheros que se van almacenando cada 10 minutos, haciendo un total de 144 por día.
- **Cloud Function:** En esta fase se inserta el código de la función a ejecutar para recopilar los datos y que los mandará al bucket mencionado anteriormente.

- **Cloud Scheduler:** Con este servicio de Google Cloud Platform se indica a la función que se ejecute cada 10 minutos. Para ello es necesario unir la función creada con el trabajo generado en el scheduler mediante un tema.

El esquema simplificado de la arquitectura que se ha seguido dentro de los servicios utilizados en Google Cloud Platform (GCP) es el siguiente:

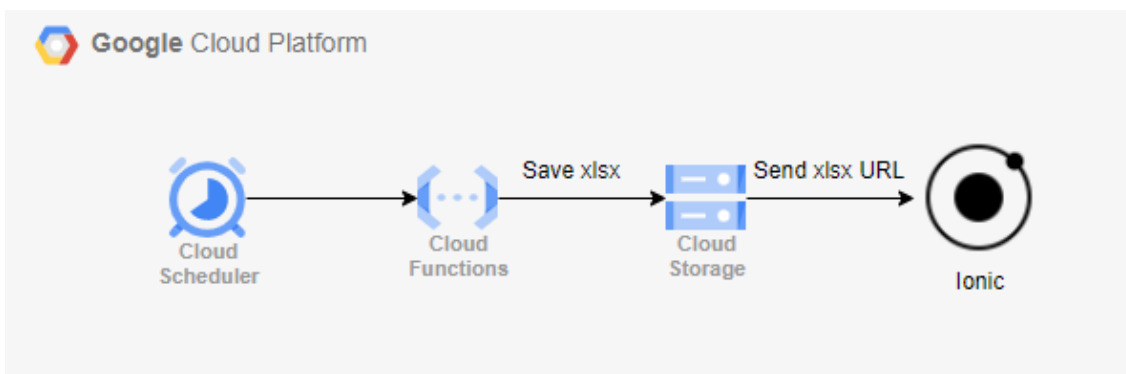


Figura 12. Esquema arquitectura GCP

En resumen, primero ha sido necesario el servicio de Cloud Storage, con el que se han almacenado los diferentes ficheros que llegan de la función encargada de hacer web scraping creada en el servidor Cloud Function y que, mediante un tema propio, está conectada a otro servicio proporcionado por Google Cloud Platform denominado Cloud Scheduler y que es el encargado de hacer que el proceso de extracción y guardado de datos se ejecute cada 10 minutos.

La configuración de cada uno de los servicios utilizados es la siguiente:

- **Cloud Storage:** La finalidad de utilizar este servicio se centra en el almacenamiento de diferentes archivos en la nube a los cuales se puede acceder desde cualquier lugar. Dentro de este servicio, se ha creado un Bucket al que se le ha puesto el nombre de traffic-sofia-bucket. La configuración para su creación se ha rellenado siguiendo los siguientes pasos:
 - **Almacenamiento de los datos:** Se ha seleccionado Bélgica (europe-west1) como organización geográfica de los datos. Además, se ha indicado que el tipo de ubicación sea de región, es decir, que tenga latencia mínima dentro de una sola región puesto que se ha pensado que con esta opción es suficiente para el tipo de datos que se está guardando.

Region
Latencia mínima dentro de una sola región

europa-west1 (Bélgica) ▼

Figura 13. Elección región en GCP

- **Clave de almacenamiento:** En este apartado se debe especificar cuánto costará almacenar los datos, realizar cualquier operación con ellos e incluso poder recuperarlos. En este caso en concreto y, por el momento, se pretende almacenar los datos a corto plazo y el acceso a ellos va a ser frecuente puesto que al querer crear un producto final que de la información en tiempo real va a tener que estar constantemente mandando peticiones a nuestro bucket para que este le de la información necesaria. Por consiguiente, la opción señalada es Standard.

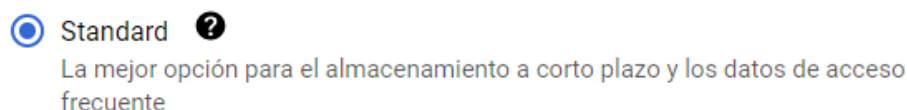


Figura 14. Elección clave de almacenamiento en GCP

- **Control de acceso:** Para acceder a los datos almacenados. En primer lugar, se debe permitir el acceso público, es decir, se va a dejar que nuestros datos sean usados para el hosting web ya que es posible que se deban poner en práctica de cara a la obtención de un producto final para el usuario. En cuanto al control de acceso, se ha señalado que lo queremos de tipo uniforme, esto quiere decir que permite el acceso uniforme al bucket usando exclusivamente permisos a nivel bucket (IAM).

Impide el acceso público

Restringe el acceso público a los datos a través de Internet. Esto evitará que el bucket se use para el hosting web. [Más información](#)

Aplicar la prevención de acceso público a este bucket

Control de acceso

- Uniforme**
Garantiza el acceso uniforme a todos los objetos del bucket mediante el uso exclusivo de permisos a nivel de bucket (IAM). Esta opción se aplicará de manera permanente después de 90 días. [Más información](#)
- Preciso**
Especifica el acceso a objetos individuales mediante el uso de permisos a nivel de objeto (LCA) además de los permisos a nivel de bucket (IAM). [Más información](#)

CONTINUAR

Figura 15. Elección control de acceso en GCP

- **Protección de los datos:** Al creer que la información que aportan los datos no es personal ni privada, puesto que la estamos cogiendo de una fuente pública y abierta a todo el mundo, se ha decidido no usar ninguna herramienta de protección, puesto que se considera que con el sistema propio que tiene predeterminado el servicio de Cloud Storage es suficiente.

Herramientas de protección

Ninguno

Figura 16. Elección protección de los datos en GCP

A rasgos generales, la configuración que se ha aplicado al servicio de Cloud Storage al crear un bucket queda de la siguiente manera:

traffic-sofia-bucket

Ubicación	Clase de almacenamiento	Acceso público	Protección
europa-west1 (Bélgica)	Standard	No público	Ninguna

Figura 17. Resumen creación bucket en GCP

Como se puede observar, el acceso sigue sin ser público y esto es necesario de cara a consumirlos fácilmente en la aplicación. Para conseguir esto se debe seguir los siguientes pasos:

- Dentro del Bucket ir a la pestaña de permisos y pulsar en añadir uno nuevo



Figura 18. Agregar un nuevo permiso en GCP Storage

- Rellenar las variables que solicita con los siguientes datos

Agregar principales a "traffic-sofia-bucket"

Agregar principales y funciones para el recurso "traffic-sofia-bucket"

Ingresa una o más principales a continuación. Después, selecciona una función para esas principales a fin de otorgarles acceso a tus recursos. Pueden asignarse varias funciones. [Más información](#)

Principales nuevas
allUsers

Rol *
Lector de objetos heredados d...
Acceso de lectura a objetos sin listado.

Condición
Condiciones inhabilitadas

+ AGREGAR OTRO ROL

GUARDAR CANCELAR

Figura 19. Tipo de permiso y parámetros a rellenar en Cloud Storage de GCP

- Se añade un nuevo permiso al listado y todos los ficheros tendrán una URL pública que permita acceder a los datos sin necesidad de iniciar sesión dentro de Google Cloud


Ubicación	Clase de almacenamiento	Acceso público	Protección
europa-west1 (Bélgica)	Standard	 Público para Internet	Ninguna

Figura 20. Resumen final creación bucket en GCP con el acceso público de los datos

En cuanto al esquema de almacenamiento de los ficheros, se va a seguir el mismo formato que el trabajado previamente en local. Existirá una carpeta por cada día en el que se recojan datos y dentro de dicha carpeta se guardarán cada uno de los ficheros con los datos recopilados a lo largo del día con un periodo de 10 minutos. El nombre de las carpetas hará referencia al día en el que se almacena el dato y el nombre del fichero a la hora y minuto en que se recoge la información. Quedando de la siguiente manera:







<input type="checkbox"/>	Nombre	Tamaño	Tipo
<input type="checkbox"/>	 2022-04-27/	—	Carpeta
<input type="checkbox"/>	 2022-04-28/	—	Carpeta
<input type="checkbox"/>	 2022-04-29/	—	Carpeta
<input type="checkbox"/>	 2022-04-30/	—	Carpeta
<input type="checkbox"/>	 2022-05-01/	—	Carpeta
<input type="checkbox"/>	 2022-05-02/	—	Carpeta

Figura 21. Orden carpetas en GCP









<input type="checkbox"/>	Nombre	Tamaño
<input type="checkbox"/>	 16-30.xlsx	436.8 KB
<input type="checkbox"/>	 16-40.xlsx	436.8 KB
<input type="checkbox"/>	 16-50.xlsx	436.8 KB
<input type="checkbox"/>	 17-00.xlsx	436.7 KB
<input type="checkbox"/>	 17-10.xlsx	436.8 KB
<input type="checkbox"/>	 17-20.xlsx	436.8 KB
<input type="checkbox"/>	 17-30.xlsx	436.6 KB
<input type="checkbox"/>	 17-40.xlsx	436.7 KB

Figura 22. Orden fichero en GCP

- **Cloud Function:** Con la utilización de dicho producto se pretende generar una función en la nube capaz de hacer web scraping. Este servicio sirve para crear aplicaciones sin la necesidad de utilizar servidores. El proceso que se ha llevado a cabo para configurar e introducir el código correspondiente a la función es el siguiente:

En primer lugar, ha sido necesario, dentro del servicio Cloud Function crear una función con la siguiente configuración:

- **Conceptos básicos:** Dentro de este apartado, se ha tenido que indicar el entorno en el que se quería trabajar. Este es 1ªgen. El nombre de la función es traffic-sofia-function y la región seleccionada vuelve a ser Bélgica (europe-west1) por ser esta la más próxima a nuestra localización.

Conceptos básicos

Entorno

1ª gen. ▼ ?

Nombre de la función *

traffic-sofia-function ?

Región

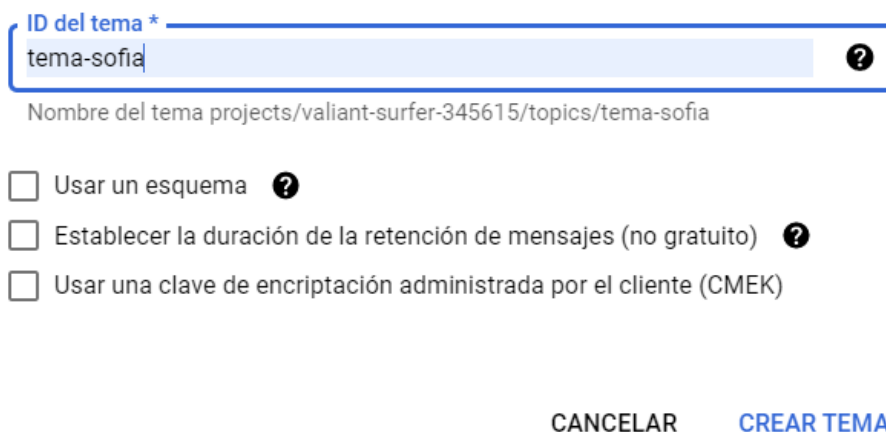
europe-west1 ▼ ?

Figura 23. Elección conceptos básicos en cloud function con GCP

- **Activador:** En este caso, se le ha indicado que el tipo de activador que queremos sea pub/sub para así poder transmitir eventos a otras herramientas de GCP, en nuestro caso en concreto, a Cloud Scheduler. La manera que tienen de conectarse ambos servicios es mediante un tema, para ello, se ha creado

uno bajo el nombre de *tema-sofia* y cuya configuración es la predeterminada en todo momento

Crear un tema



ID del tema *
tema-sofia

Nombre del tema projects/valiant-surfer-345615/topics/tema-sofia

Usar un esquema

Establecer la duración de la retención de mensajes (no gratuito)

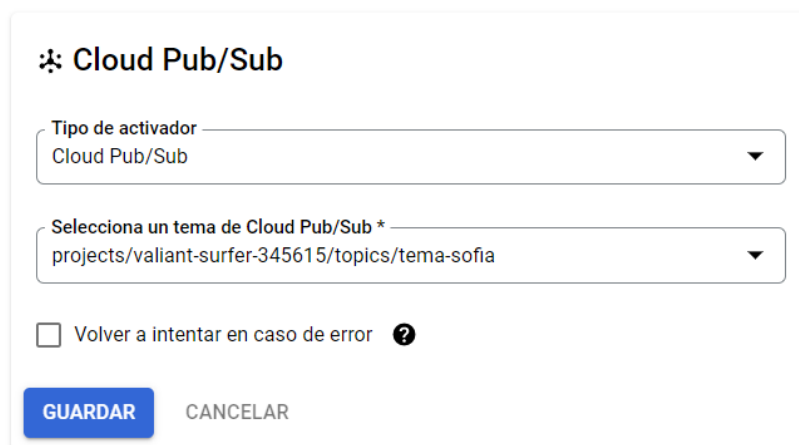
Usar una clave de encriptación administrada por el cliente (CMEK)

CANCELAR CREAR TEMA

Figura 24. Elección tema del activador en cloud function con GCP

Por lo tanto, la configuración del activador para nuestra función quedaría de la siguiente manera:

Activador



⚙️ Cloud Pub/Sub

Tipo de activador
Cloud Pub/Sub

Selecciona un tema de Cloud Pub/Sub *
projects/valiant-surfer-345615/topics/tema-sofia

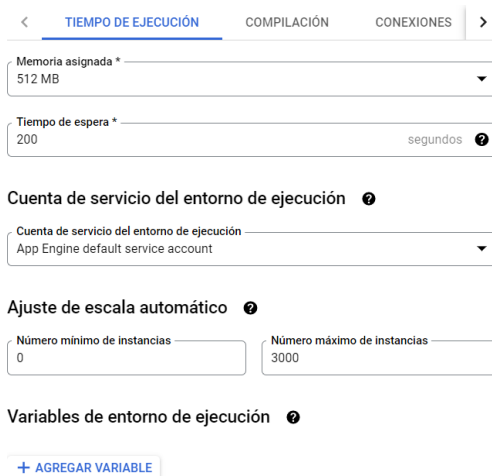
Volver a intentar en caso de error

GUARDAR CANCELAR

Figura 25. Elección activador en cloud function con GCP

- **Configuración del entorno de ejecución:** En este apartado se debe indicar la memoria asignada, los segundos de espera antes de finalizar la ejecución del Cloud Function, la cuenta de servicio que la función asume como su identidad, el número mínimo y máximo de instancias de respuesta y las variables de entorno. En este caso, se ha marcado 512 MB como memoria asignada, 200 segundos de tiempo de espera, la cuenta de servicio de entorno de ejecución que venía por defecto, un intervalo de instancias entre 0 y 3000 y no se ha

creado ninguna variable de entorno de ejecución. Por lo tanto, la configuración queda de la siguiente manera:



< TIEMPO DE EJECUCIÓN COMPILACIÓN CONEXIONES >

Memoria asignada *
512 MB

Tiempo de espera *
200 segundos ?

Cuenta de servicio del entorno de ejecución ?
Cuenta de servicio del entorno de ejecución
App Engine default service account

Ajuste de escala automático ?
Número mínimo de instancias: 0
Número máximo de instancias: 3000

Variables de entorno de ejecución ?
[+ AGREGAR VARIABLE](#)

Figura 26. Elección configuración entorno de ejecución en cloud function con GCP

- **Configuración de la compilación:** Se han dejado los valores como estaban por defecto, ya que se ha decidido que no se quería aplicar este servicio puesto que no se ha entendido como necesario.
- **Configuración de las conexiones:** En este apartado existe distinción entre la configuración de entrada y salida. En el caso de la entrada se ha permitido todo el tráfico (valor predeterminado de la configuración), en el caso de la salida, de nuevo se han dejado los valores predeterminados, es decir, no hemos aplicado ningún conector de VPC.



< DE EJECUCIÓN COMPILACIÓN CONEXIONES SEGUI >

Configuración de entrada ?
 Permitir todo el tráfico
 Permitir solo el tráfico interno
Solo se permite el tráfico desde las redes de VPC en el mismo proyecto o el mismo perímetro de VPC SC.
 Permitir el tráfico interno y el proveniente de Cloud Load Balancing
Se permite el tráfico desde las redes de VPC en el mismo proyecto, el mismo perímetro de VPC SC o desde Cloud Load Balancing.

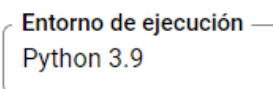
Configuración de salida ?
Tu función puede enviar solicitudes a Internet de forma predeterminada, pero no a los recursos en las redes de VPC. Para enviar solicitudes a los recursos en tu red de VPC, crea o selecciona un conector de VPC que ya se haya creado en la misma región que la función.
Conector de VPC
Ninguno
[Crear un conector de VPC sin servidores](#)
 Enrutar solo las solicitudes a las IP privadas mediante el conector de VPC
 Enrutar todo el tráfico mediante el conector de VPC

Figura 27. Elección configuración conexiones en cloud function con GCP

- **Configuración de seguridad:** Se ha decidido no hacer uso de este servicio que ofrece Cloud Function ya que los datos que se almacenan no contienen información sensible.

En cuanto a la parte de código, se debe tener en cuenta el entorno de ejecución, el punto de entrada y el código fuente de la función junto a sus librerías. Los pasos a seguir para configurar este apartado son los siguiente:

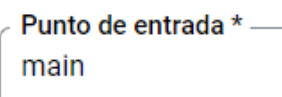
- **Entorno de ejecución:** Al estar trabajando con python, se debe indicar también la versión. En nuestro caso, será Python 3.9.



Entorno de ejecución —
Python 3.9

Figura 28. Elección entorno de ejecución en cloud function con GCP

- **Punto de entrada:** Se debe indicar el nombre del fichero que se quiere ejecutar. Por ejemplo, en este proyecto, la parte de código va a estar compuesta por una función cuya finalidad es hacer web scraping y un main que llama a dicha función por lo que, el punto de entrada o lo que se tiene que ejecutar para que funcione todo el código es el main.



Punto de entrada * —
main

Figura 29. Elección entorno de ejecución en cloud function con GCP

- **Código:** Este apartado está dividido en tres partes diferentes, cada una de ellas con una función y un código distinto. Para el código se ha partido de la función ya creada y ejecutada en local, función a la que se le ha tenido que aplicar diversos cambios para que funcione en la nube.

Los tres archivos que forman la función son los siguientes:

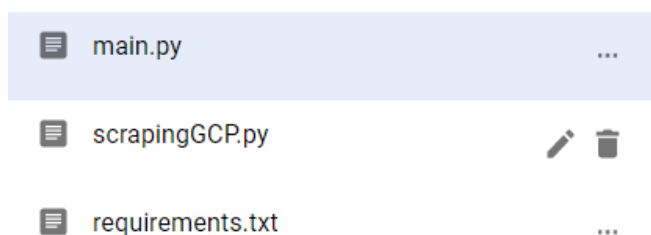


Figura 30. Archivos que componen la función del cloud function con GCP

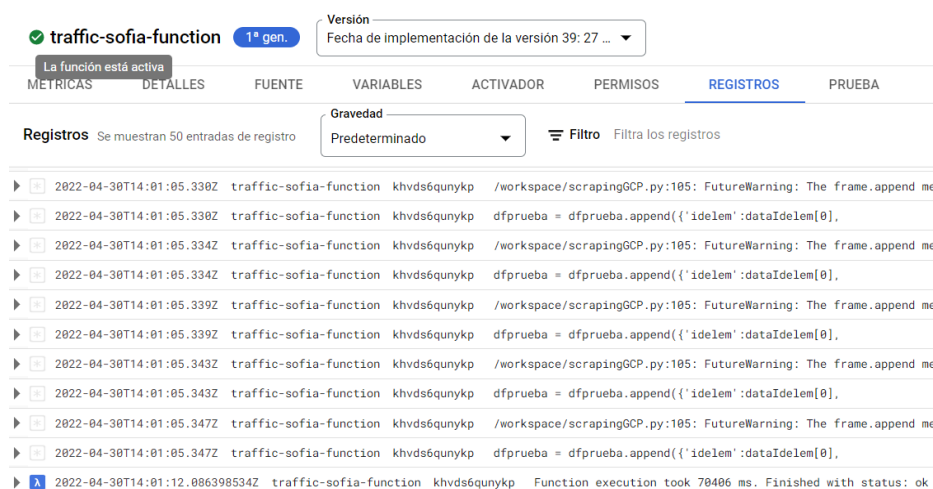
En el fichero scrapingGCP.py es donde está la función que realiza el web scraping. El único cambio que se ha hecho en cuanto a la función generada en local es la dirección sobre la que se van a guardar cada uno de los ficheros. Anteriormente, en local, estos se almacenaban en el propio sistema del ordenador, ahora se pretende que esto se haga en el bucket creado previamente.

Dentro del fichero main.py encontramos la llamada a la función anterior. Para ello, se debe tener en cuenta que, a diferencia del código ejecutado en local, aquí será necesario importar la llamada a dicha función.

Por último, dentro de requirements.txt se debe indicar todas las librerías o dependencias que se tienen que instalar para que funcione el código. Además, existe la opción de indicar la versión que se quiere. En este caso, no ha sido necesario indicar la versión. Las dependencias necesarias para el código son las siguientes:

- **pandas**: Librería de Python destinada a manejar y analizar estructuras de datos.
- **numpy**: Librería de Python centrada en el cálculo numérico y análisis de datos principalmente cuando se trabaja con grandes volúmenes de datos.
- **requests**: Librería de Python que ayuda a realizar peticiones HTTP.
- **lxml**: Biblioteca de Python que ayuda a manejar fácilmente archivos XML y HTML. Está destinada a trabajos con web scraping.
- **beautifulsoup4**: Librería de Python que extrae información del contenido en HTML o XML.
- **fsspec**: Es una dependencia de GCP para leer y escribir datos en el storage.
- **gcsfs**: Interfaz de Google Cloud Storage que depende de fsspec
- **openpyxl**: Librería de Python destinada a leer o escribir el formato de Excel.

El último paso una vez rellenada la configuración y el apartado de código es implementar la función. Tras generarla existe un apartado de prueba en el que se puede ejecutar la función y ver su funcionamiento. Además, observar el comportamiento paso a paso de la función en el apartado de registros y los diferentes errores que esta arroja.



The screenshot shows the 'REGISTROS' (Logs) tab for the 'traffic-sofia-function'. The logs display a series of execution events with timestamps, source names, and error messages. The errors are 'FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use DataFrame.append instead.'

Timestamp	Source	Message
2022-04-30T14:01:05.330Z	traffic-sofia-function	/workspace/scrapingGCP.py:105: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use DataFrame.append instead.
2022-04-30T14:01:05.330Z	traffic-sofia-function	dfprueba = dfprueba.append({'idelem':dataIdelem[0],
2022-04-30T14:01:05.334Z	traffic-sofia-function	/workspace/scrapingGCP.py:105: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use DataFrame.append instead.
2022-04-30T14:01:05.334Z	traffic-sofia-function	dfprueba = dfprueba.append({'idelem':dataIdelem[0],
2022-04-30T14:01:05.339Z	traffic-sofia-function	/workspace/scrapingGCP.py:105: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use DataFrame.append instead.
2022-04-30T14:01:05.339Z	traffic-sofia-function	dfprueba = dfprueba.append({'idelem':dataIdelem[0],
2022-04-30T14:01:05.343Z	traffic-sofia-function	/workspace/scrapingGCP.py:105: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use DataFrame.append instead.
2022-04-30T14:01:05.343Z	traffic-sofia-function	dfprueba = dfprueba.append({'idelem':dataIdelem[0],
2022-04-30T14:01:05.347Z	traffic-sofia-function	/workspace/scrapingGCP.py:105: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use DataFrame.append instead.
2022-04-30T14:01:05.347Z	traffic-sofia-function	dfprueba = dfprueba.append({'idelem':dataIdelem[0],
2022-04-30T14:01:12.086398534Z	traffic-sofia-function	Function execution took 70406 ms. Finished with status: ok

Figura 31. Registros de la función en cloud function con GCP

- **Cloud Scheduler:** Es un servicio proporcionado por Google Cloud Platform para programar trabajos. En el caso, se ha utilizado para llamar y ejecutar la función de obtención de los datos cada 10 minutos. Para la configuración se debe tener en cuenta:
 - **Definición del programa:** En este apartado, aparte de rellenar la región (Bélgica), el nombre o zona horaria, es importante destacar la frecuencia. Aquí es donde se debe indicar cada cuanto tiempo se quiere que este haga la llamada a la función. El formato que sigue este apartado es el siguiente:

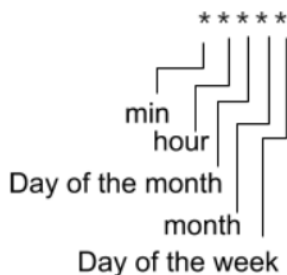


Figura 32. Instrucciones definir el programa en cloud scheduler con GCP

Por lo que, en nuestro caso, al querer ejecutarla durante todo el día cada diez minutos. Se pone lo siguiente:

`* / 10 * * * *`

La configuración total de este apartado queda de la siguiente manera:

- **Define el programa**

Nombre *
Ejecute-function-traffic
Debe ser único en los trabajos de la misma región.

Región *
europe-west1 (Bélgica)

Descripción

Frecuencia *
*/10 * * * *

Las programaciones se especifican con el formato cron de Unix. P. ej., cada minuto: * * * * *, cada 3 horas: * * * */3 * * * *, todos los lunes a las 9:00 a.m.: * * * 9 * * 1*. [Más información](#)

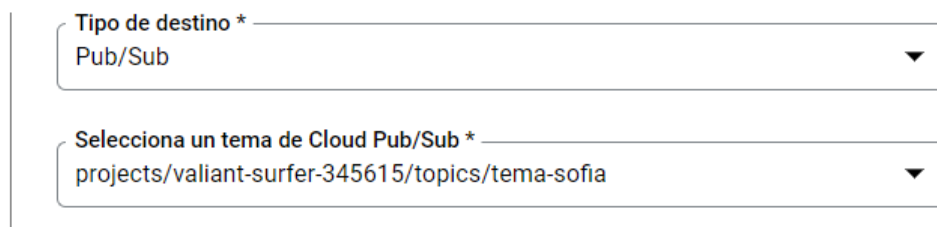
Zona horaria *
hora de verano de Europa central (000)

Los trabajos establecidos en zonas horarias afectadas por el horario de verano pueden ejecutarse fuera de la cadencia durante el cambio a DST. Si usas una zona horaria ITC, puedes evitar el problema. [Más información](#)

Figura 33. configuración total definición del programa en cloud scheduler con GCP

- **Configuración de la ejecución:** En este apartado es donde se debe unir la función y el programa que se está creando. Para ello, se va a utilizar el tema creado anteriormente y el activador pub/sub. La información rellena en este apartado queda de la siguiente manera:

- **Configura la ejecución**



The screenshot shows two dropdown menus for configuring a Cloud Scheduler job. The first dropdown is labeled 'Tipo de destino *' and has 'Pub/Sub' selected. The second dropdown is labeled 'Selecciona un tema de Cloud Pub/Sub *' and has 'projects/valiant-surfer-345615/topics/tema-sofia' selected.

Figura 34. configuración ejecución en cloud scheduler con GCP

- **Configuración de reintentos:** En este apartado se ha indicado que, en caso de existir algún error no se reintente el trabajo de nuevo. Además, como la ejecución de la función requiere de un tiempo mínimo aproximado de un minuto y, en algunas ocasiones tardaba varios minutos, (sobre todo esto sucedía cuando se ejecutaba la función en local) para evitar problemas, se decidió poner un tiempo máximo de duración de 10 minutos para así asegurar que no llegue al límite de tiempo la función. En caso de que falle una ejecución, se ha establecido un intervalo entre 5 segundos y 1 hora para volver a reintentar el trabajo, tiempo suficiente para modificar cualquier error surgido. Por último, se ha rellena el valor de las duplicidades máximas como 5 y, por consiguiente, el tiempo entre reintentos será de 10, el doble que las duplicidades máximas. La configuración final queda de la siguiente manera:

- **Establece los parámetros de configuración opcionales**

Configuración de reintentos

Si un trabajo no se completa de forma correcta, se reintentará con una retirada exponencial según los parámetros de la configuración de reintentos. [Más información](#)



The screenshot shows the 'Configuración de reintentos' section with the following settings:

- Cantidad máx. de reintentos:** 0 (Cantidad máxima de reintentos para un trabajo con errores)
- Duración máx. del reintento *:** 10m (Límite de tiempo para reintentar un trabajo con errores (0 s significa ilimitado))
- Duración mín. de la retirada *:** 5s (Tiempo mínimo de espera antes de reintentar un trabajo después de que falla)
- Duración máx. de la retirada *:** 1h (Tiempo máximo de espera antes de reintentar un trabajo después de que falla)
- Duplicaciones máximas *:** 5 (El tiempo entre reintentos será el doble del de las duplicaciones máximas)

Figura 35. configuración de reintentos en cloud scheduler con GCP

Una vez que está todo el proceso configurado, se ejecutará sin necesidad de tener que intervenir para su ejecución.

Preprocesamiento de los datos:

La parte más importante de todo proyecto, se enfoca en los datos con los que se va a trabajar. Tener un buen conjunto de datos ayuda a obtener mejores resultados en el modelo y también, a poder sacar grandes conclusiones que ayuden a conseguir el objetivo. Antes de comprobar si los datos son veraces, se ha procedido a realizar una serie de modificaciones previas con el objetivo de poder mejorar el conjunto de datos. También, se ha realizado un análisis genérico sobre su obtención mediante la plataforma que nos ofrece Google en la nube.

En primer lugar, se ha revisado que toda la configuración realizada en GCP para la obtención de los datos en tiempo real, ha funcionado correctamente. Para ello, ha sido necesaria la ayuda que ofrece la propia plataforma y sobre la que se han realizado las siguientes comprobaciones:

- **La función se ejecuta cada 10 minutos:** Verificando este apartado se comprueba no solo que el trabajo creado en Cloud Scheduler esté bien configurado y se ejecute correctamente cada diez minutos. Sino que, también se comprueba que la llamada a la función creada en Cloud Function está correctamente configurada. Para su verificación se pueden observar las siguientes gráficas en las que se puede ver las invocaciones a la función que hay a lo largo de las 12 últimas horas:



Figura 36. Gráfica ejecución de la función última hora GCP

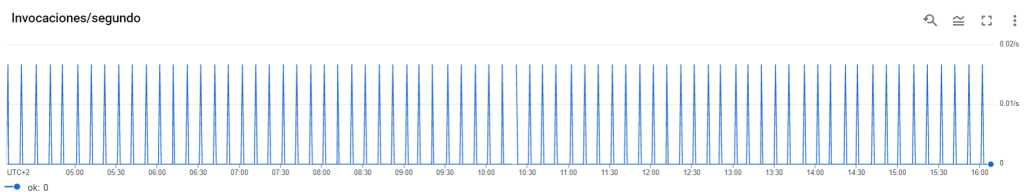


Figura 37. Gráfica ejecución de la función 12 últimas horas GCP

Como se puede comprobar, las llamadas siguen una periodicidad en todo momento, es decir, todo funciona correctamente y se llama a la función cada 10 minutos. No existe ninguna anomalía al respecto.

- **No existe ningún error en la función:** Para comprobar este apartado, es necesario acceder a los logs de la función. En ellos se observa como no hay ninguno bajo el título de advertencia, error, crítico, alerta o emergencia. Esto quiere decir que todo lo que se le está pidiendo a la función se ejecuta sin ningún problema, es decir, todos los ficheros almacenan las variables correctamente con un valor en su interior. No obstante, se debe tener en cuenta que, aunque la función se ejecute correctamente y almacene

valores, estos pueden no ser correctos. Este proceso de análisis de la veracidad de los resultados se llevará a cabo más adelante.

- Los ficheros se almacenan correctamente:** En este apartado se debe mirar si existe un fichero cada diez minutos con datos en su interior y que se guarden en la dirección correcta. Es decir, existe una carpeta diferente en el bucket para guardar los datos particionados. Se debe comprobar que, el paso de los días se hace correcto en cuanto al guardado de los ficheros se refiere. Para ello, se debe acceder al bucket y realizar las siguientes comprobaciones:
 - Hay 144 ficheros por día:** Dentro de cada una de las carpetas que se dividen en días tiene que existir un total de 144 ficheros debido a que el día tiene un total de 24 horas con 60 minutos cada una de ellas. Al ejecutar la función cada 10 minutos, a lo largo del día se ejecuta un total de 144 veces, resultado obtenido al multiplicar las horas del día por los minutos que tiene cada hora, es decir, $24 \times 60 = 144$ ficheros.

Filas por página: 50 ▼ 1 – 50 de 144 < >

Figura 38. Número de ficheros almacenados por día en GCP

- Comprobamos que el fichero tiene datos:** Para dicha comprobación, será necesario revisar el tamaño del fichero. Pese a que no todos los ficheros son iguales, existe una similitud de tamaños entre todos ellos que verifica que en su interior las variables tienen datos, pero esto no nos asegura que sean los correctos en ningún momento, se deberá comprobar posteriormente.

Nombre	Tamaño
00-00.xlsx	427.6 KB
00-10.xlsx	427.6 KB
00-20.xlsx	427.4 KB
00-30.xlsx	427.3 KB
00-40.xlsx	351.1 KB
00-50.xlsx	426.9 KB
01-00.xlsx	426.6 KB
01-10.xlsx	426.9 KB
01-20.xlsx	426.8 KB

Figura 39. Ficheros y memoria almacenados en GCP

Preparación y análisis de los datos:

Una vez que se sabe que el proceso de almacenar el conjunto de datos está correcto, es decir, tras comprobar que se han guardado todos los ficheros con sus variables correspondientes y la información de cada una de ellas en su interior, se debe verificar que los datos proporcionados por cada una de las variables tienen sentido realizando un análisis más detallado de las variables. Para poder cargar el conjunto de datos en Power BI y, teniendo en cuenta las restricciones que se tiene al usar la versión gratuita, fue necesario realizar una serie de funciones previas usando de nuevo el servicio de Cloud Function que nos ofrece GCP, con el objetivo de reunir todo el conjunto de datos de dos semanas en un mismo fichero.

- **Pasos previos al análisis de datos:** En este apartado se explicarán las diferentes funciones generadas para unificar el conjunto de datos y los problemas encontrados a lo largo del proceso. Pensando en el modelo, su optimización de cara a la aplicación y en el análisis de datos que se quería obtener, se realizaron dos funciones diferentes. Los datos que se deseaba unificar en un primer momento eran los recogidos durante dos semanas consecutivas y, las funciones creadas han sido:
 - **Función que engloba los datos por día de la semana:** En este apartado se unifican todos los valores de los diferentes días de la semana que están almacenados en los ficheros por separado. En este caso en concreto, se trabaja con los datos desde el día 2 de mayo de 2022 hasta el 15 de mayo de 2022, ambos días incluidos. Esta aplicación crea un total de siete ficheros distintos, uno por cada día de la semana.

En cuanto a la configuración de la función, ha sido bastante similar a la mencionada anteriormente, a excepción del valor de la memoria que se ha tenido que introducir el máximo permitido, 8GB, y aún así nos hemos encontrado con problemas puesto que se hacían demasiadas solicitudes al bucket y esto colapsaba. La manera de solucionarlo ha sido particionado, se han creado dos ficheros por cada día con la unión de la mitad de los datos y posteriormente se han juntado obteniendo como resultado final un solo fichero por cada semana.

<input type="checkbox"/>	Nombre	Tamaño
<input type="checkbox"/>	 Domingo.csv	188.7 MB
<input type="checkbox"/>	 Jueves.csv	187.2 MB
<input type="checkbox"/>	 Lunes.csv	192.3 MB
<input type="checkbox"/>	 Martes.csv	191.6 MB
<input type="checkbox"/>	 Miércoles.csv	191.9 MB
<input type="checkbox"/>	 Sabado.csv	188.3 MB
<input type="checkbox"/>	 Viernes.csv	191.1 MB

Figura 40. Ficheros con la unión de los datos de cada día de la semana en GCP Storage durante quince días (2-5-22/15-5-22)

- **Función que unifica todos los datos de dos semanas:** Una vez ejecutada la función mencionada anteriormente y, al tener un total de 7 ficheros, el siguiente paso es unificar todos ellos en uno solo.

El proceso en este caso de creación es igual que en el caso anterior a excepción del código. El problema encontrado en este caso es que se supera el límite de filas que excel permite. Para solucionar esto, fue necesario modificar el tipo de fichero que se iba a almacenar cambiando el formato de excel a CSV.



Figura 41. Fichero con la unión de todos datos recogidos en GCP Storage durante quince días (2-5-22/15-5-22)

- **Análisis de los datos:** Con los datos unificados para realizar el análisis es momento de cargarlos en una herramienta de visualización e intentar obtener alguna conclusión con ellos, es decir, analizarlos. El estudio se ha centrado principalmente en dos de las variables que más información aportan y las que más vamos a utilizar en el modelo y las visualizaciones posteriormente. Estas son la ocupación y velocidad, con los datos almacenados desde el 2 de mayo hasta el 15 del mismo mes se ha podido sacar las siguientes conclusiones:

- **Variable ocupación:** En este caso, se muestra un diagrama de caja-bigote del conjunto de datos total, en él se puede observar como la mayor distribución de los datos la tenemos entre los valores 66,45 y 105,87. Valores bastante elevados en general

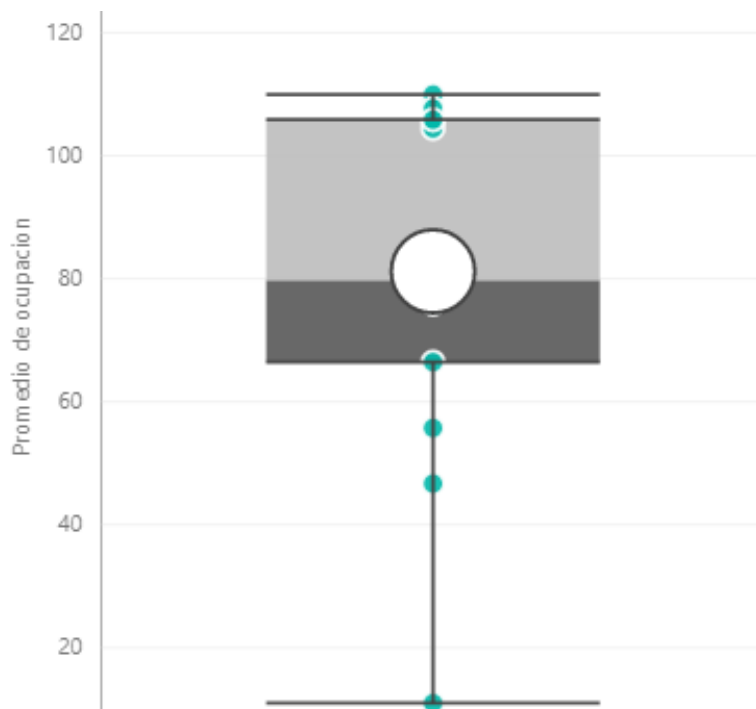


Figura 42. Diagrama de caja-bigote con la distribución de los datos de la ocupación del tráfico

Median Type	Inclusive
Whisker Type	Min/Max
Mean	81,20
Quartile 1	66,45
Median	79,76
Quartile 3	105,87
Maximum	109,96
Minimum	10,99
IQR	39,43
Upper Whisker	109,96
Lower Whisker	10,99

Figura 43. Datos detallados de la distribución de la variable ocupación

Si se entra en detalle en los diferentes días de la semana para comparar la ocupación de las carreteras en relación al día de la semana y, teniendo en cuenta que el lunes día dos de mayo fue festivo en Madrid, el promedio de la ocupación de cada uno de los días representado mediante una tarjeta en Power BI es el siguiente:

- **Lunes:** En este caso, vemos como la ocupación para el día dos de mayo es inferior a la del lunes siguiente. Lo más probable es que esto sea debido a que el lunes día dos de mayo es fiesta en Madrid y la gente no tiene que trasladarse a los colegios ni al trabajo y, por lo tanto, disminuye la ocupación. No obstante, esta no se asemeja a un sábado debido a que se debe tener en cuenta el regreso del puente y por ello es más similar el valor a la media que se obtiene un domingo.

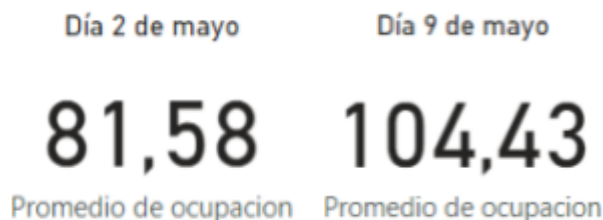


Figura 44. Promedio de la variable ocupación de los lunes día dos y nueve de mayo

- **Martes:** Como se puede observar, en este caso los valores son bastante similares. Es importante mencionar que no hay gran diferencia en el día 3 de mayo, pese a haber sido fiesta el día de antes. Esto puede deberse a diferentes razones, algunas de ellas son: que la gente se movió en el puente por Madrid y apenas salieron fuera de la capital, los que salieron de puente volvieron el lunes y, por eso hay tanto promedio de ocupación de las carreteras ese día o también puede ser debido a que no todo el mundo que salió fuera de Madrid en el puente regresó el martes sino que lo fue haciendo escalonadamente a lo largo de la semana.



Figura 45. Promedio de la variable ocupación de los martes día tres y diez de mayo

- **Miércoles:** Aquí existe una gran diferencia entre ambos días, es algo raro. Una de las posibles causas es que, al coger únicamente los valores que los sensores han sido capaces de medir bien, es decir, eliminar aquellos datos que nos devuelven números negativos, se haya disminuido en gran cantidad los datos recopilados y no se obtenga un valor coherente en función a los datos reales. No obstante, se puede decir que el miércoles día 4 hay una mayor ocupación porque regresó gente que todavía no lo había hecho del puente.



Figura 46. Promedio de la variable ocupación de los miércoles día cuatro y once de mayo

- **Jueves:** En este apartado sucede algo similar al anterior, no es muy común que el promedio en las carreteras de un día con jornada laboral y colegios esté en torno al 11. No obstante, se puede observar con el valor del día 12 de mayo que apenas hay una gran ocupación ese día.



Figura 47. Promedio de la variable ocupación de los jueves día cinco y doce de mayo

- **Viernes:** En cuanto a los resultados de la ocupación media de los viernes podemos observar una pequeña diferencia entre ambos días. Además, se puede ver como no hay una ocupación mayor a la de cualquier otro día pese a ser viernes.



Figura 48. Promedio de la variable ocupación de los viernes día seis y trece de mayo

→ **Sábado:** Como es común, los fines de semana hay menos movimiento en las carreteras y, por consiguiente, la ocupación disminuye dando los siguientes resultados:



Figura 49. Promedio de la variable ocupación de los sábados día siete y catorce de mayo

→ **Domingo:** De la misma manera que los sábados, el tráfico en las carreteras es inferior al de un día normal, pero superior al de los sábados pues también se debe añadir el regreso del fin de semana para aquellas personas que hayan decidido salir de la capital.



Figura 50. Promedio de la variable ocupación de los domingos día ocho y quince de mayo

En resumen, se puede observar cómo, a excepción del resultado obtenido para el jueves día cinco de mayo, los valores son los esperados en todo momento. Los fines de semana y días festivos hay un menor movimiento en las carreteras y menos ocupación que entre semana, siendo los martes los días de mayor ocupación. En estos resultados se debe tener en cuenta la flexibilidad del teletrabajo y, por consiguiente, la posibilidad de empeorar el patrón que se genera durante la semana ya que no todas las empresas van todas las semanas a la oficina, ni lo hacen en el mismo día de la semana. La siguiente gráfica representa una forma de ver de manera general y en un mismo gráfico la ocupación del tráfico en las carreteras de Madrid en función a los días:

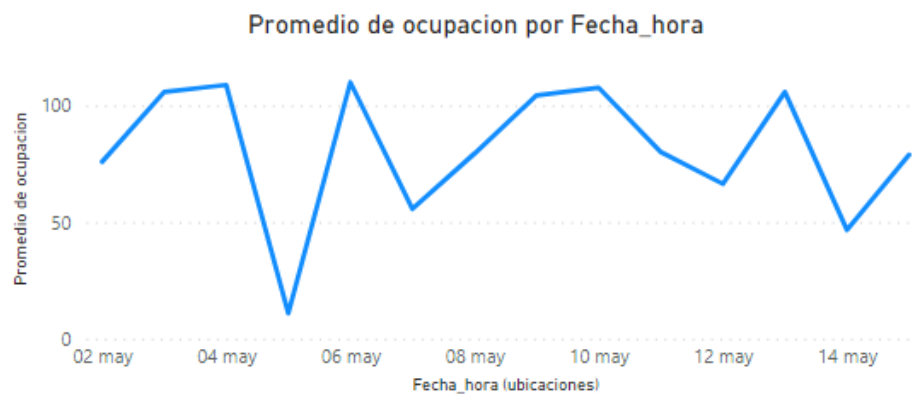


Figura 51. Gráfica de líneas que representa el promedio de la variable ocupación por días

- **Variable velocidad:** Con el diagrama de caja-bigote de todo el conjunto de datos almacenados durante dos semanas y con la representación del promedio de ellos, se puede observar como existe una mayor concentración de valores y, apenas se salen de los cuartiles.

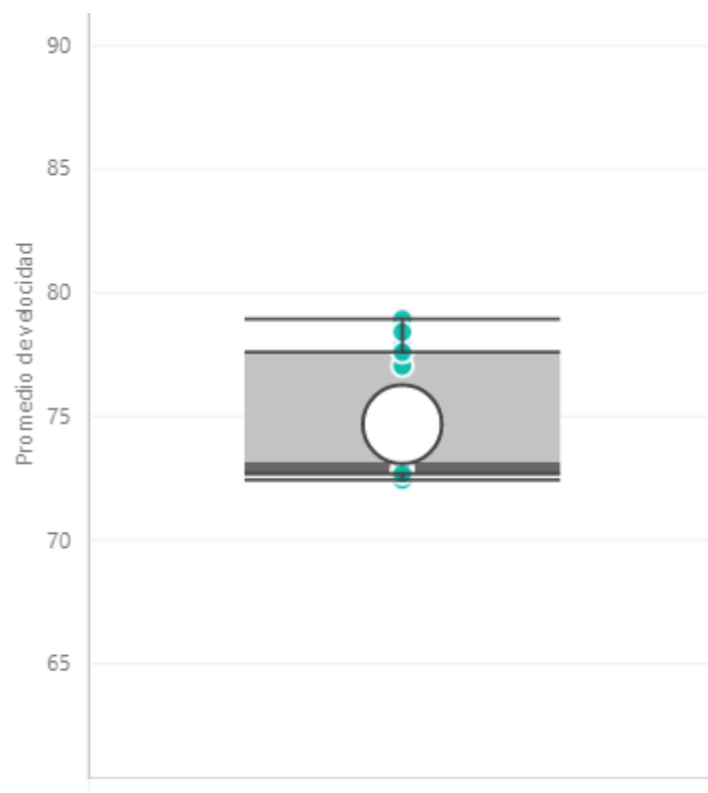


Figura 52. Diagrama de caja-bigote con la distribución de los datos de la velocidad del tráfico

Median Type	Inclusive
Whisker Type	Min/Max
Mean	74,68
Quartile 1	72,69
Median	73,16
Quartile 3	77,60
Maximum	78,94
Minimum	72,43
IQR	4,91
Upper Whisker	78,94
Lower Whisker	72,43

Figura 53. Datos detallados de la distribución de la variable velocidad

En cuanto a la distribución del promedio de la velocidad en los diferentes días, se puede ver una linealidad entre ambas semanas, siendo los fines de semana los valores más elevados y dejando los días de entre semana con velocidades inferiores. También, se observa cómo el lunes día 2 de mayo la velocidad sigue siendo superior a la de otro lunes normal (9 de mayo). En la siguiente gráfica se puede observar visualmente lo mencionado:

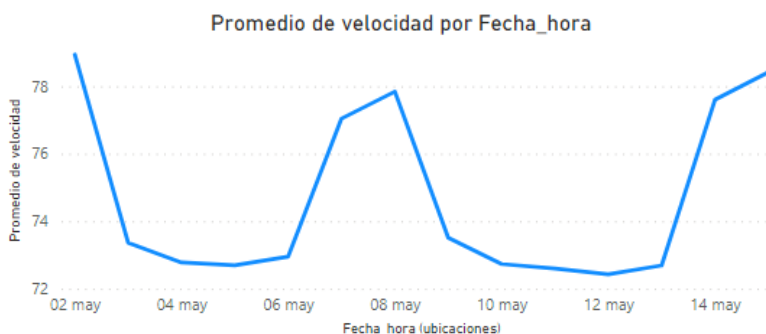


Figura 54. Gráfica de líneas que representa el promedio de la variable velocidad por días

También, es importante hacer énfasis en un dato curioso, el valor máximo de velocidad registrado es de 209 Kilómetros por hora, casi el doble de lo permitido por las carreteras de la capital. Este valor ha sido registrado el jueves día cinco de mayo. Siguiendo a este dato, se puede observar un registro del día 11 de mayo de una velocidad de 208 kilómetros por hora.

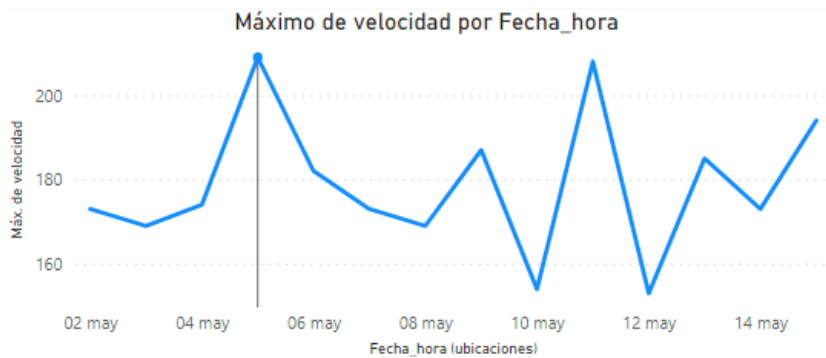


Figura 55. Gráfica de líneas que representa los valores máximos recogidos de la variable velocidad por días

Siguiendo con los excesos de velocidad, se puede observar como durante dos semanas ha habido un total de 1055 datos en los que la velocidad ha sido superior a 120 kilómetros por hora

recuento datos
superiores a 120km/h

1055

Recuento de velocidad

Figura 56. Etiqueta que muestra el número de datos recogidos con una velocidad superior a 120 km/h

También, se puede ver como estos datos ocurren mayoritariamente en zonas horarias con pocos atascos y, siguiendo una periodicidad.



Figura 57. Gráfica de líneas que representa la distribución de fecha y hora de los datos de velocidad superiores a 120 km/h

- **Relación entre variables:** Teniendo en cuenta las variables velocidad, ocupación e intensidad, se han sacado las siguientes conclusiones entre variables:
 - **Variable ocupación e intensidad:** Existe una relación directamente proporcional entre ambas variables, a mayor ocupación en la carretera, mayor intensidad. La única diferencia es la escala de cada una de las variables, ya que los valores de ocupación son mucho más pequeños que los de intensidad y no se podía representar las gráficas unificadas. A continuación, se puede observar la distribución de los valores en día y hora de la semana y se puede comprobar como hay una similitud para todos los días a excepción del jueves día 5 de mayo que, como hemos mencionado anteriormente, es una excepción donde los datos carecen de sentido.

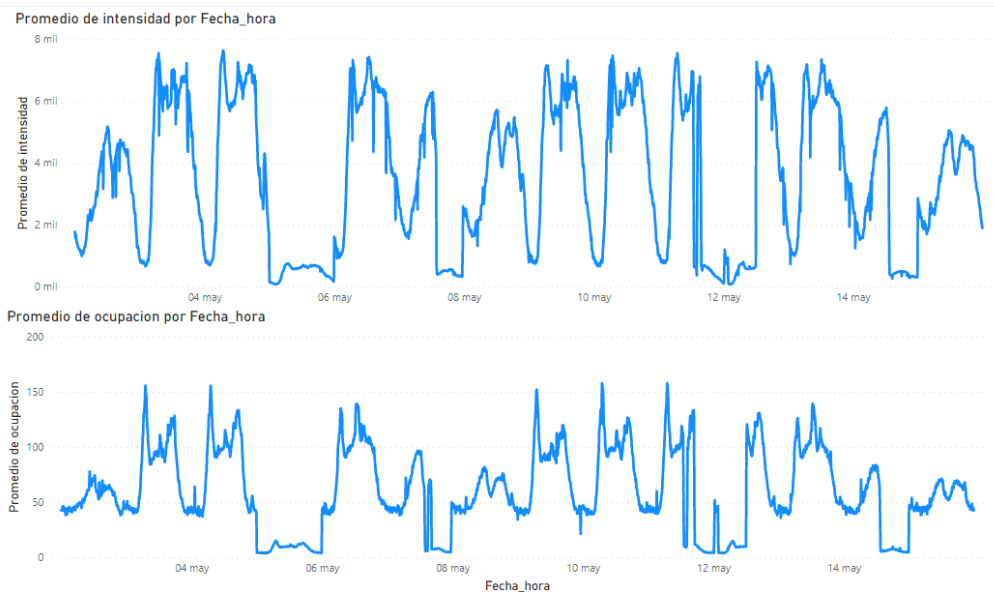


Figura 58. Gráficas de líneas que representa la relación que hay entre la variable intensidad y ocupación en función al tiempo

→ **Variable ocupación e intensidad con velocidad:** Por norma general, la relación que deberían tener estas variables es inversamente proporcional, cuanto mayor sea la ocupación e intensidad de las carreteras, a menor velocidad se circulará por ellas. Pues mediante las gráficas se puede observar cómo no existe esta relación.

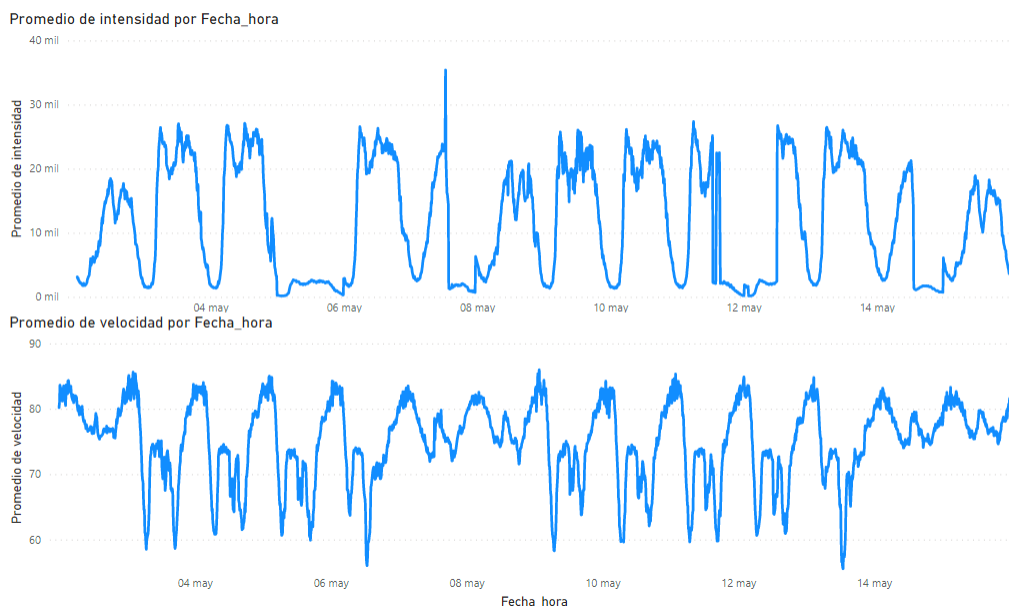


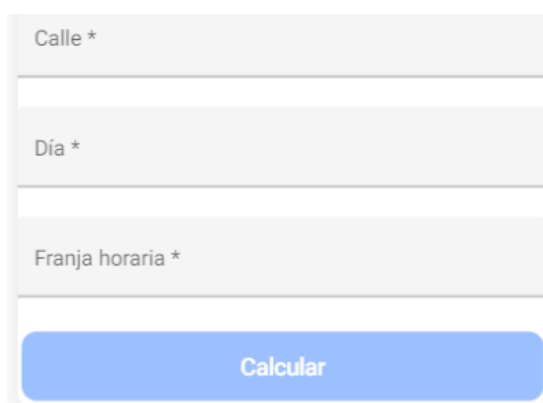
Figura 59. Gráficas de líneas que representa la relación que hay entre la variable intensidad y velocidad en función al tiempo

Modelo:

Tras conseguir el conjunto de datos que nos aporta información sobre la congestión de tráfico en Madrid es momento de generar un modelo de regresión en el que pretendemos ayudar al usuario a recomendarle la mejor hora para viajar a un lugar con la finalidad principal de evitar las horas más concurridas y, por consiguiente, intentar distribuir el tráfico a lo largo del día y no centrarse en dos puntos concretos, siendo estos principalmente el comienzo del día con la entrada al trabajo y a los colegios y a sus respectivas salidas. Todo este modelo ha sido implementado directamente desde la aplicación. Por lo tanto, para darle esta solución al usuario, se ha necesitado solicitar ciertos datos mediante la creación de un formulario que pida al usuario la información básica necesaria para ejecutar el modelo. Esta información es la siguiente:

- **Dirección de la calle a la que va a ir:** En este apartado aparecerá un listado de la localización de los diferentes sensores urbanos para seleccionar. Es importante tener en cuenta que, si existe más de un sensor en una misma calle, se deberá seleccionar el que se encuentre más cerca del lugar del destino al que el usuario quiera ir.
- **Día de la semana en el que quiere viajar:** En esta sección aparecerá un listado de siete opciones diferentes: lunes, martes, miércoles, jueves, viernes, sábado o domingo. El usuario deberá seleccionar en el formulario el día que quiere viajar ya que el comportamiento del tráfico depende en mayor medida del día en el que se viaje.
- **Franja horaria:** Para no dar como resultado una hora que no le sea útil al usuario, se ha decidido solicitar también la franja horaria sobre la cual quiere saber el resultado. En esta parte del formulario habrá 4 opciones diferentes. Estas son:
 - de 10:00 h a 14:00 h
 - de 14:00 h a 18:00 h
 - de 18:00 h a 00:00 h
 - de 00:00 h a 10:00 h

El formulario con las tres variables a solicitar tiene el siguiente aspecto en la aplicación:



The image shows a web form with three input fields stacked vertically. The first field is labeled 'Calle *', the second 'Día *', and the third 'Franja horaria *'. Below these fields is a blue button with the text 'Calcular' in white.

Figura 60. Imagen formulario a rellenar en la página web para calcular el modelo de predicción

Una vez recopilada la información de las variables del formulario rellenas y, para poder darle un resultado, es momento de pasar a generar el modelo. En este caso, el primer paso que se

debe realizar es la aplicación de diferentes filtros al conjunto de datos total con el que se va a trabajar con dos finalidades principales como son:

- **Disminuir el tamaño de nuestro conjunto de datos:** Al final el modelo se está calculando en local y, tener un conjunto de datos muy grande supone que los procesos tarden mucho en ejecutarse y esto no sea viable de cara al tiempo de respuesta con el que se va a trabajar.
- **Eliminar datos que empeoren el modelo:** Cuando se filtra el conjunto de datos teniendo en cuenta los valores introducidos en el cuestionario previo, se eliminan datos que no tienen un comportamiento similar al solicitado por el usuario, es decir, teniendo en cuenta que el tráfico se comporta diferente en función al día de la semana, si se entrena el modelo con esos datos el resultado que tendremos no se asemejará con el verdadero. Lo mismo pasa con la localización, la congestión del tráfico depende de si esa carretera está más o menos congestionada y, por consiguiente lo idóneo será tener en cuenta únicamente dicha calle. En definitiva, se pretende eliminar aquellos datos que únicamente aportan ruido al resultado del modelo.

Como el objetivo del modelo es recomendar al usuario la hora idónea para ir hacia esa calle, nos encontramos ante un modelo de regresión debido a que, lo que queremos predecir son valores continuos que se encuentran en un intervalo entre las 00:00 y las 23:50 horas del día.

Para implementar este modelo es necesario el uso de expresiones matemáticas que nos ayuden a predecir el horario o momento de tiempo idóneo para viajar. Esta expresión es encontrar el valor mínimo de ocupación que cumpla con los filtros seleccionados en el cuestionario.

Al crear el modelo directamente en la aplicación, no ha sido necesario la implementación posterior del código con la app. De esta manera, se ahorra tiempo de ejecución.

En añadido, se ha generado un modelo en Python de series temporales en el que se han utilizado técnicas propias de aprendizaje automático, en específico, redes neuronales de memoria a corto plazo (LSTM). La finalidad principal es predecir la intensidad del tráfico en un futuro. Además, se han implementado diferentes gráficas que ayudan a ver los resultados del modelo.

Para realizar este modelo será necesario filtrar en un primer momento por la calle que se quiere estudiar y así, generar una matriz con una única dimensión. En función a ello y a tener un dataset con únicamente la variable fecha y la intensidad como variable que queremos predecir y, que debe ser estar en formato numérico. Se procede a dividir el conjunto de datos total en dos, uno para el entrenamiento y otro para el test posterior, con esto dividido se comienza a realizar los pasos previos a aplicar el modelo y a su implementación.

Finalmente, se procede a evaluar los resultados obtenidos calculando la raíz del error cuadrático medio (RMSE) y comparando los resultados obtenidos, este resultado indica la cantidad de errores que hay entre dos conjuntos de datos. Se debe tener en cuenta que el número de épocas usadas para entrenar el modelo únicamente ha sido una para que así el kernel siguiera funcionando. Los resultados obtenidos en una primera prueba son más

positivos de los esperados, ya que se trabaja con grandes conjuntos de datos y un entrenamiento con pocas épocas.

Train Score: 193.13 RMSE
Test Score: 122.56 RMSE

Figura 61. Resultados del RMSE obtenido usando redes neuronales a corto plazo

El resultado obtenido por el modelo representado mediante una gráfica en la que se compara el conjunto de datos para entrenar, la predicción obtenida en dicho conjunto de datos y la predicción obtenida en el test es el siguiente

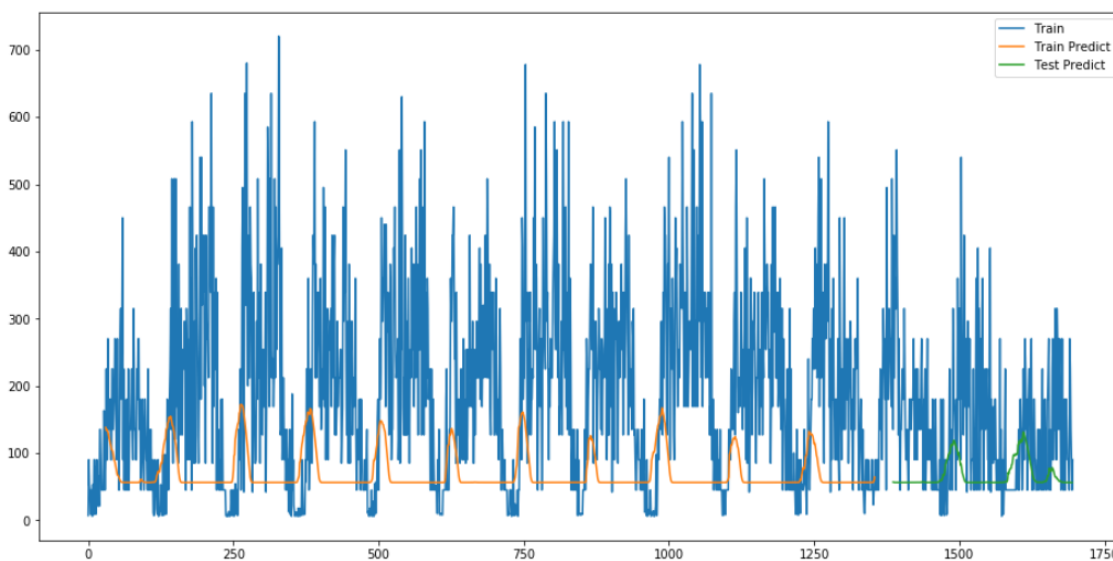


Figura 62. Gráfico de líneas que representa el conjunto de datos de entrenamiento y las predicciones de ese conjunto y del de test

Visualización:

En este ámbito, la herramienta por excelencia empleada ha sido Power BI. Con ella, se han realizado las gráficas y cálculos matemáticos que ayudan a ofrecer un resultado visual sobre el análisis del conjunto de datos mencionado anteriormente cuya finalidad principal era poder obtener conclusiones. También, se han implementado diferentes visualizaciones con la misma herramienta Power BI en la aplicación, que se explicarán más en detalle, pero que pretenden dar una visión general, a largo plazo, del tráfico de las carreteras en el municipio de Madrid.

Por último, dentro del ámbito de la visualización se ha hecho uso de código para representar un mapa integrado en la aplicación. Lo que refleja el mapa es, dadas unas coordenadas y, teniendo en cuenta la variable que refleja la ocupación, se ha añadido a dicho mapa una burbuja por cada uno de los sensores medidores de los datos que están dibujados de un color u otro dependiendo de la ocupación en tiempo real. Los colores seleccionados serán de color verde para representar que el tráfico de las carreteras no está muy congestionado, amarillo en caso de un tramo intermedio y rojo como advertencia de tráfico lento o mucha ocupación en la zona.

Para realizar este trabajo ha sido necesario realizar unos pasos previos importantes de cara a obtener unas mejores visualizaciones y que la información que aporte sea útil. Estos son:

- **Realización de un brainstorming:** Una vez que conocemos el conjunto de datos, se deberán apuntar todas las ideas que se tienen o curiosidades que surjan acerca de los datos, es decir, en el caso en concreto del proyecto una de las ideas que se ha apuntado ha sido la de intentar reflejar si existe algún patrón con el tráfico de los datos en función al día de la semana. Estas ideas nos ayudan a generar las gráficas.

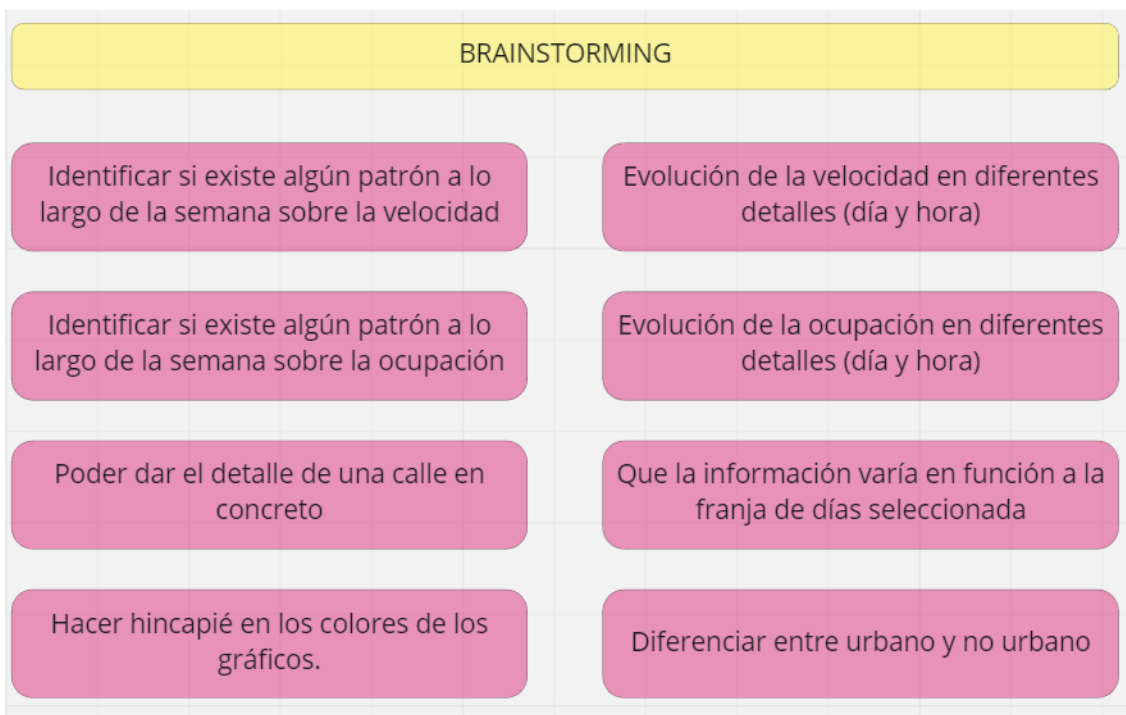


Figura 63. Brainstorming de ideas para los gráficos de visualización

- **Storytelling:** Una vez enumeradas diferentes ideas de visualización y antes de pasar a prototipar el diseño de los diferentes cuadros de mandos, es necesario tener en cuenta que, la mejor manera de organizar la información es enfocarse en la creación de una historia, es decir, debemos contar una historia con los datos que tenemos y, para ello, es recomendable ir de gráficas más genéricas a otras más concretas. En este caso en concreto, el Storytelling que se quiere realizar refleja en primer lugar información general sobre la ocupación o velocidad de las carreteras por días. Posteriormente, se centra más en el detalle en horas y minutos.
- **Crear un prototipo:** Tras tener las ideas, es hora de organizar todo en un prototipo indicando los KPIs o filtros que se quieren usar, que se deseen representar en cada sección y el gráfico más idóneo para su representación y entendimiento. El prototipo generado consta de varias páginas:

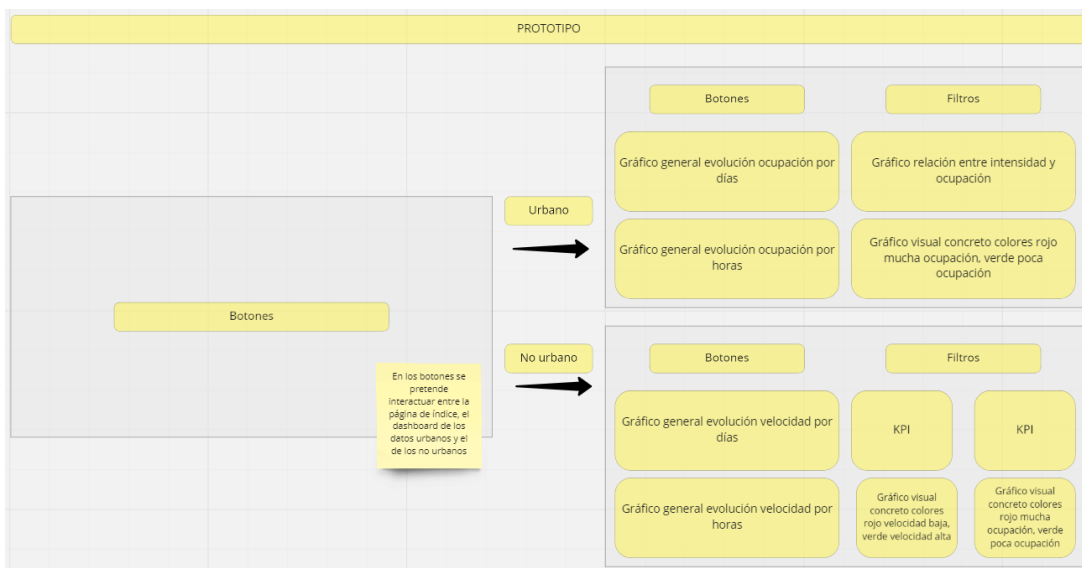


Figura 64. Diseño prototipo de las vistas de visualización

- **Implementar este prototipo en la herramienta en cuestión:** En este caso, Power BI y el código propio de la aplicación para el mapa.

A modo de conclusión, cabe destacar que los dos frentes en los que se ha enfocado la visualización han sido el análisis de datos y el producto final. De todo lo realizado se deben recalcar las siguientes actividades:

- **Uso de herramientas de visualización para el análisis de los datos:** En este apartado se han generado principalmente cálculos matemáticos como puede ser la media, para comprobar los datos y se ha representado la información mediante diferentes tarjetas o KPIs de Power BI. Además, se ha comprobado la distribución de los datos. Todos los gráficos con resultados interesantes para el análisis han sido adjuntados en el apartado de análisis de los datos explicado anteriormente. No obstante, se adjunta un ejemplo de un cuadro de mandos realizado sobre la variable velocidad:

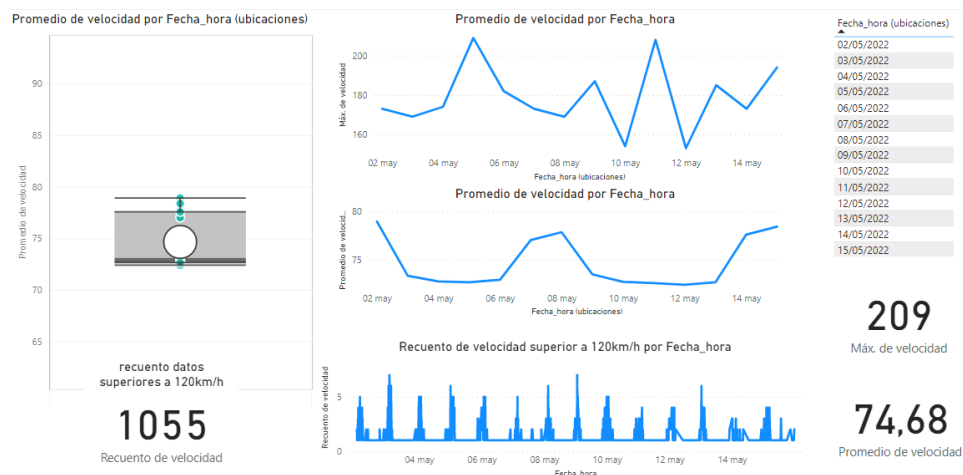


Figura 65. Ejemplo visualizaciones empleadas en el análisis de datos usando Power BI

- **Implementación del cuadro de mandos creado en Power BI con la aplicación:** Este apartado ha sido el más complejo de la visualización, en él se ha implementado el prototipo creado para las visualizaciones de los datos. El resultado obtenido son tres páginas diferentes relacionadas entre sí para poder interactuar. Las diferentes visualizaciones son las siguientes:

- **Índice:** Este apartado únicamente está compuesto de dos botones (urbano y no urbano) unidos cada uno de ellos con su página correspondiente de datos y visualizaciones creada. Además se ha añadido como fondo una imagen relacionada con el tema en cuestión. El resultado queda de la siguiente manera:



Figura 66. Imagen índice de las visualizaciones

- **Página con datos del tráfico urbano:** En este apartado se ha replicado el prototipo generado, compuesto por diferentes gráficos de líneas, barras y burbujas. Varios filtros para obtener información detallada sobre un día o varios en concreto o la calle exacta sobre la que se quiere conocer la información. Además, se han añadido dos botones para acceder al cuadro de mandos con los datos no urbanos y otro para ir hacia atrás. De igual manera que en la página anterior se ha incluido una imagen de fondo quedando como resultado final el siguiente dashboard:

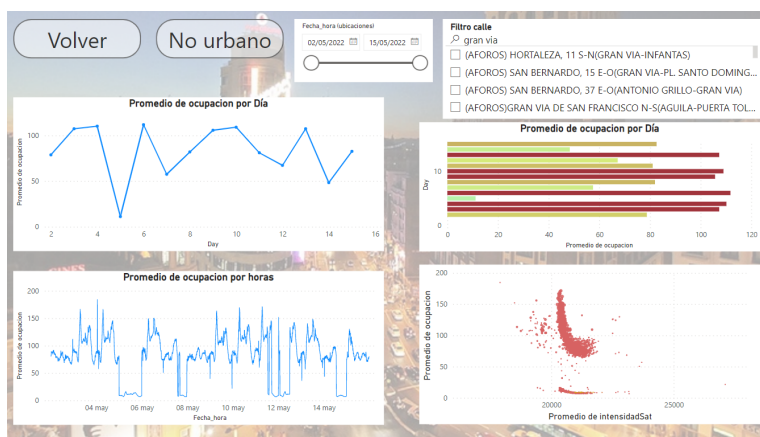


Figura 67. Imagen cuadro de mandos datos urbanos

- **Vista con datos del tráfico no urbano:** Esta vista contiene implementadas todas las opciones mencionadas en el prototipo, gráficas, filtros y botones. También, se le ha añadido una imagen de fondo quedando el resultado final de la siguiente manera:

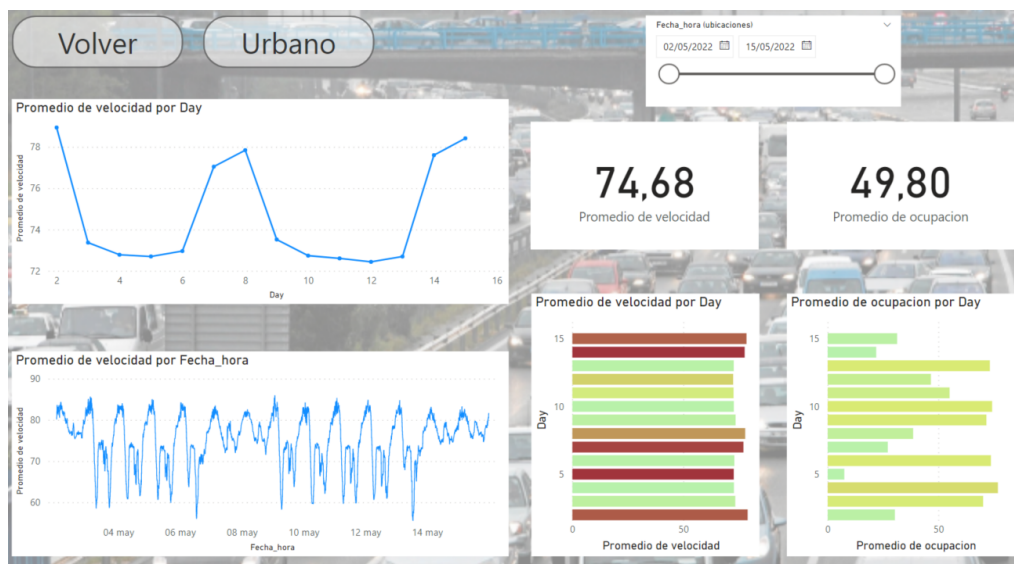


Figura 68. Imágen cuadro de mandos datos no urbanos

Las diferentes pantallas también han sido colocadas para ser leídas por la app descargada en un dispositivo móvil. Tras tener las visualizaciones creadas, se tenían que implementar en la aplicación, para este apartado se tuvo una serie de problemas relacionados con las diferentes versiones de pago que Power BI ofrece. Para poder publicar los resultados en la web siendo esta privada era necesario tener un paquete superior, en caso de querer hacerlo público se requería permisos del administrador y estos tras solicitarlos nunca llegaron. No obstante, existe una opción de publicarlo sin ningún coste adicional para la comunidad de la cuenta personal, es decir, para todos los miembros que con el link inicien sesión haciendo uso del correo de la Universidad Europea. Tras verificar una posible solución, se implementa en la aplicación un botón que al acceder a él lleva al usuario a una página de Power BI para, tras iniciar sesión, poder ver las visualizaciones. Es importante mencionar que, si se elige la opción de abrirlo desde la web el formato en el que se verán será como de ordenador pero, sin embargo, desde el teléfono móvil, al abrirlo desde la APP se verá con formato móvil.

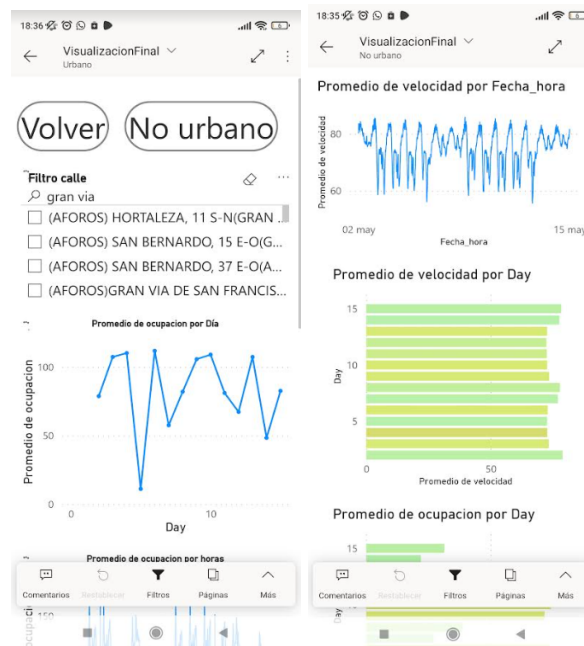


Figura 69. Imágenes visualizaciones cuadros de mandos en formato para el dispositivo móvil

- **Realización del mapa en la aplicación:** Este apartado es más técnico y enfocado a la parte de desarrollo de código. De toda la implementación cabe destacar la necesidad de cambiar el formato de los datos de ubicación de cada sensor de UTM a longitud y latitud. Además, se ha necesitado de la creación de condiciones *if* y *else* para los colores de cada sensor en el mapa. El resultado obtenido va a ir cambiando en tiempo real. Por lo que al codificarlo, ha sido necesario ver la hora a la que nos encontramos para acceder a esos datos. A continuación, se muestran dos imágenes de diferentes horas del día donde se puede observar la diferencia que hay en la ocupación de cada hora:

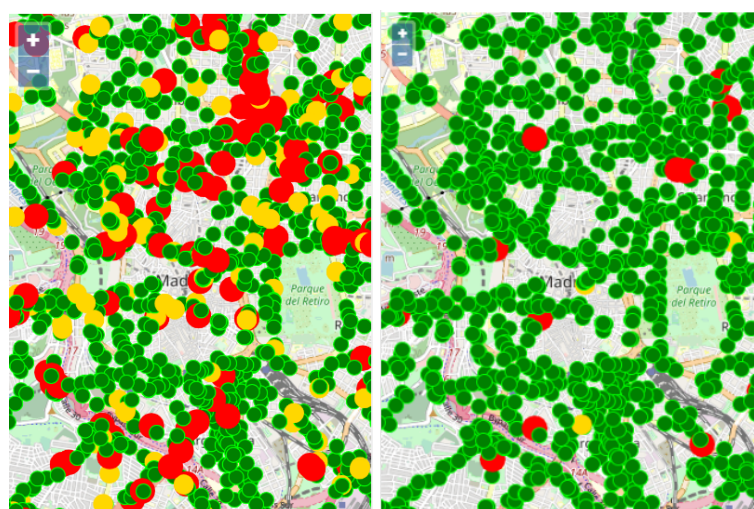


Figura 70. Imágenes visualizaciones mapa implementado en la aplicación que representa la ocupación de tráfico en madrid en dos horas distintas

Producto final / Aplicación:

Tras desarrollar todo el proceso de recopilación de los datos, realización de un modelo, diferentes visualizaciones y analítica, es necesario implementarlo en un producto final de cara a los usuarios. En este caso en concreto, se ha decidido desarrollar una aplicación puesto que se considera que es la mejor herramienta para utilizar. Al final, el usuario la va a usar principalmente cuando vaya a hacer un viaje y, lo más probable es que cuando consulte la aplicación se encuentre fuera de casa por lo que la realización de una página web, que era la otra opción, nos resultaba menos útil.

Una vez que se había determinado cuál iba a ser el producto final, es decir, una aplicación, se necesitaba saber, en primer lugar, que es lo que se quería representar, qué información se quiere enseñar a nuestros usuarios que sirva para tener un mayor conocimiento del tráfico en Madrid. Realizando un pequeño brainstorming de ideas surgen las siguientes opciones de información para enseñar al cliente:

- **Enseñarle el tráfico en las carreteras de Madrid en tiempo real:** Esto ayudará a conocer si van a coger en este momento el coche, las zonas que a evitar, e incluso aquellas en las que se recomienda tener más precaución puesto que el tráfico será más lento o incluso si le es viable circular en ese momento.
- **Más información sobre el tráfico de Madrid:** Con este apartado, se pretende presentar al cliente la evolución del tráfico en Madrid ya no solo en tiempo real si no enfocado más a poder analizar en general el tráfico. Para este apartado se deben usar los datos de diferentes fechas y realizar diferentes gráficos de visualización.
- **Ayudar al cliente a recomendar la mejor hora para viajar un día determinado:** Con esto el cliente podrá ver la hora más óptima para viajar a una calle en concreto indicada por el usuario. Este es un caso de uso muy común. Por ejemplo, necesito ir al centro de Madrid al médico y al solicitar la cita me aparecen diferentes opciones de hora para el mismo día y, a mi me es indiferente la hora. Con la información que se le puede ofrecer al usuario en este apartado, se podrá determinar la mejor hora para seleccionar su cita y así evitar atascos y consumir menos combustible.

En segundo lugar, debemos estudiar cómo ordenar la información que se quiere representar en la aplicación y de qué manera. Para ello, se ha realizado un prototipo explicando la idea que se desea llegar a obtener en la aplicación. Este prototipo está compuesto por un total de 5 pantallas, dos de ellas enfocadas en los pasos previos de registro y login de la aplicación y las otras tres, son las pantallas que se encuentran cuando se accede a la aplicación. Los prototipos son los siguientes:

- **Pantallas de registro y login:** Estas pantallas están conectadas entre sí para poder navegar entre ellas. Además, deben estar conectadas a una base de datos con la finalidad de almacenar los datos de los clientes que, o bien accedan a la aplicación por primera vez (registro), o bien quiera iniciar sesión ya que ya había sido registrado previamente y cuyos datos están ya almacenados en la base de datos.

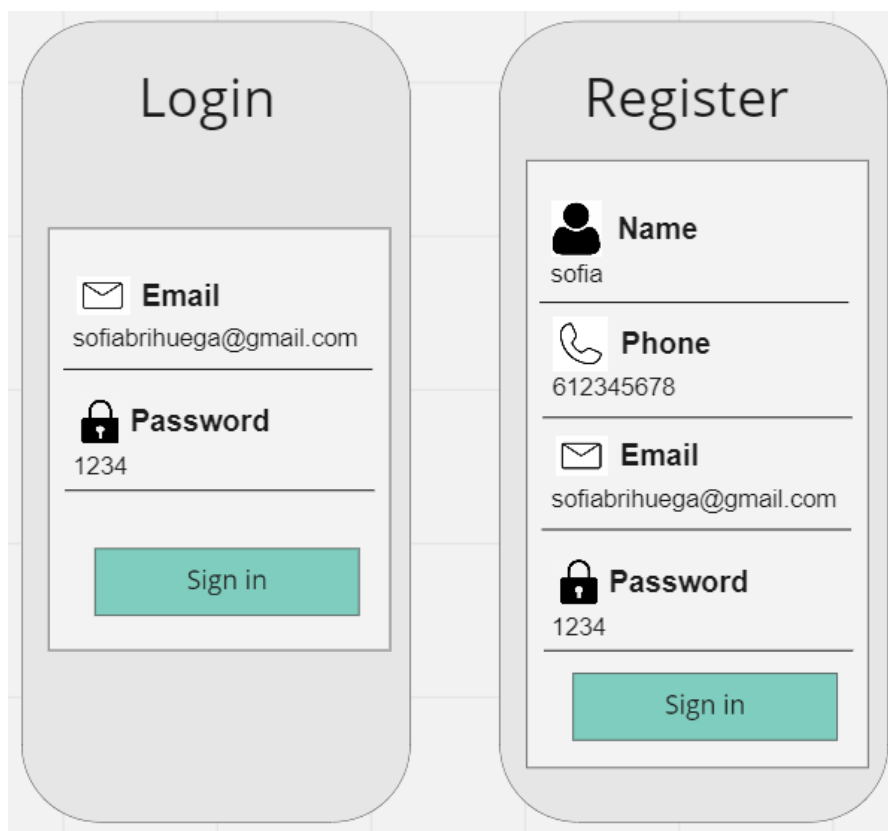


Figura 71. Prototipo de las pestañas de registro y login para la aplicación

- **Pantallas del interior de la aplicación:** Una vez realizado el registro y login, al acceder al interior de la aplicación, se puede ver como esta está dividida en los tres grandes puntos de ideas principales mencionados anteriormente. Cada una de las páginas tiene un formato diferente.

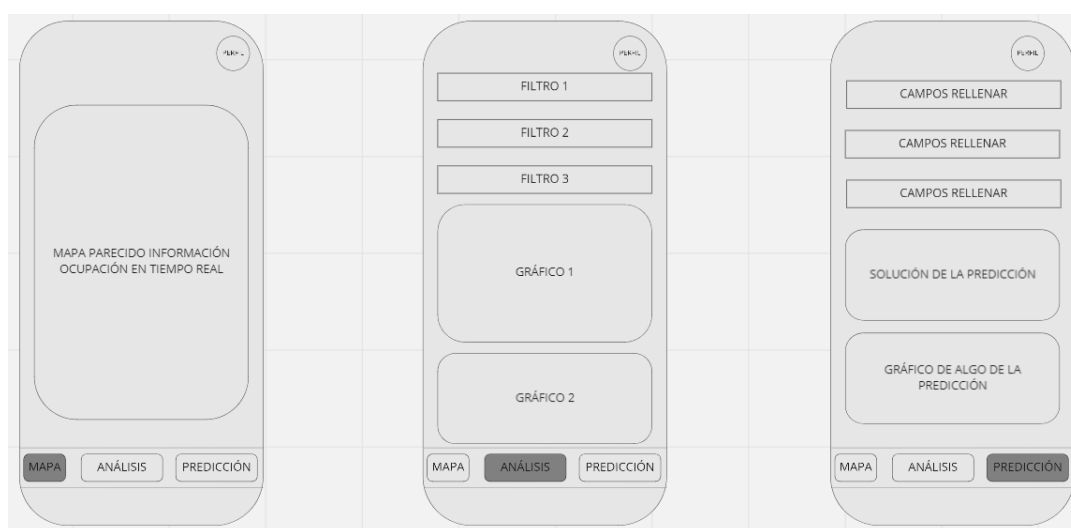


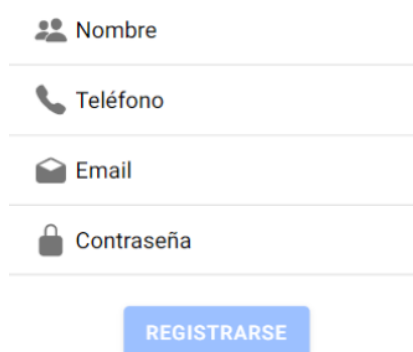
Figura 72. Prototipo de las páginas que deben salir al entrar en la aplicación

Sofía Corral Caballero

Con el prototipo e idea diseñados, se procede a implementar. Para ello, se ha usado ionic, un SDK de código abierto para aplicaciones híbridas. La decisión de usar ionic viene enfocada en la optimización del tiempo, ya que de esta manera con una única codificación se puede abrir la aplicación en dispositivos móviles con distintos sistemas operativos (Android e ios).

La aplicación una vez generada la base, se puede dividir en dos grandes grupos. El primero de ellos es la realización de las pantallas de registro y login del usuario y la unión de esta con una base de datos que almacene cada usuario. Los pasos realizados en este apartado son los siguientes:

- Crear el cuestionario con los datos que se van a solicitar al usuario rellenar tanto en la pantalla de registro como la de login

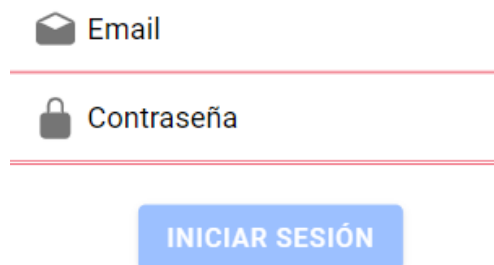


Formulario de registro con los siguientes campos:

- Nombre
- Teléfono
- Email
- Contraseña

Botón: **REGISTRARSE**

Figura 73. Imagen formulario a rellenar de la aplicación para crear una cuenta



Formulario de inicio de sesión con los siguientes campos:

- Email
- Contraseña

Botón: **INICIAR SESIÓN**

Figura 74. Imagen formulario a rellenar de la aplicación para iniciar sesión

- Implementar la interactividad entre páginas. En este apartado se tendrá en cuenta la unión necesaria entre ambas páginas y que, al rellenar los datos para entrar a la aplicación tras unas comprobaciones previas, esta redirija a la primera pantalla de la aplicación.

¿Tienes ya una cuenta? [Iniciar sesión](#)

¿No tiene cuenta? [Registrarse](#)

Figura 75. Imagen integración entre las pantallas de iniciar sesión y registrarse

Sofía Corral Caballero

- Creación de una base de datos con el uso de la herramienta MySQL workbench. La base de datos creada tendrá como registros los valores necesarios a rellenar en la pantalla de registro.

id	nombre	password	telefono	email
1	Sofia	1234	6543	sofia@prueba.com

Figura 76. Imagen variables y un ejemplo de la base de datos creada en MySQL para almacenar los usuarios de la aplicación

- Unificar la base de datos con la aplicación, es decir, implementarla para almacenar los diferentes usuarios que acceden a la aplicación. Para la realización de este paso ha sido necesario ayudarse de Postman, herramienta sobre la que se han realizado diferentes consultas.
- Modificar el estilo de las pantallas a través del css de cada una de estas con la finalidad de añadir una imagen de fondo o emoticonos en los registros entre otros.

Como valor añadido a estas páginas se ha querido no dejar al usuario entrar a la aplicación hasta que no haya rellenado todos los datos correctamente, es decir, si se le solicita un correo electrónico será obligatorio que el campo a rellenar tenga un '@' y un '.' seguido de cualquier combinación de caracteres. Hasta que no estén rellenados todos los campos correctamente, no se activará el botón de registro o login.

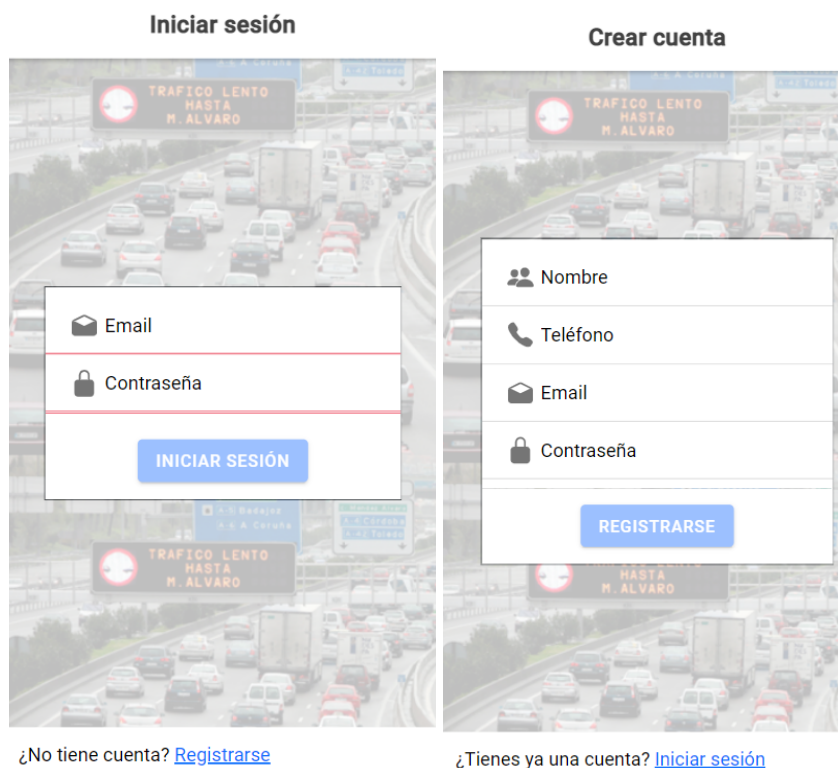


Figura 77. Imágenes formularios para crear una cuenta y registrarse sin rellenar los datos

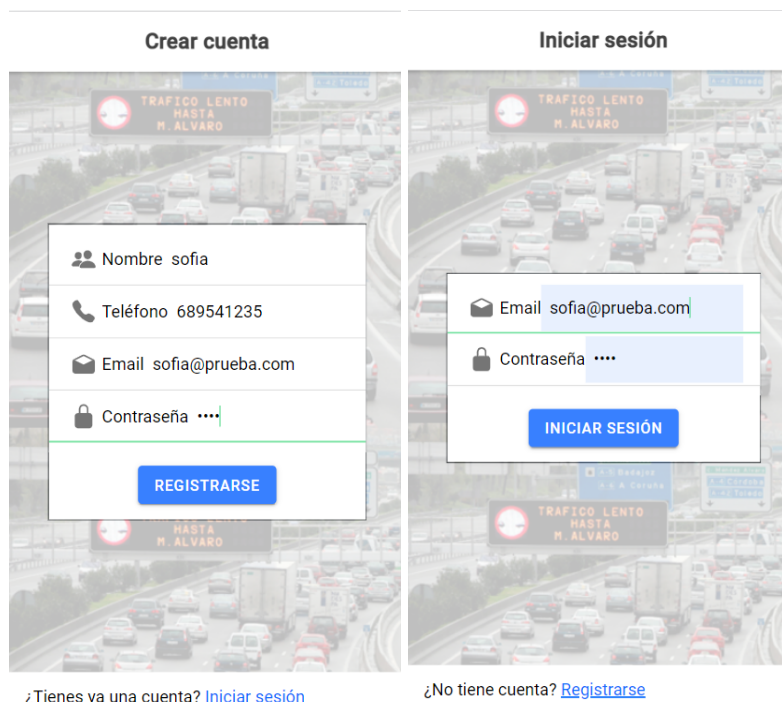


Figura 78. Imágenes formularios para crear una cuenta y registrarse con los datos rellenos y el botón activado

El segundo gran bloque para desarrollar en la aplicación se corresponde con la creación de cada una de las pantallas internas. Los diferentes pasos que se han desarrollado en este apartado son los siguientes:

- **Implementación del mapa:** Este apartado es el que se quiere representar en la primera pantalla de la aplicación. Para este proceso ha sido necesario cargar los datos que se recogen en el bucket del tráfico de Madrid y realizar una pequeñas modificación en el propio código de la aplicación, puesto que la variable que se indica la localización de cada uno de los sensores está escrita en formato UTM y, para representar el mapa ha sido necesario cambiarlo a longitud y latitud. Además, se han tenido que codificar los puntos que aparecerán en la misma localización que los sensores e indicar el color de estos en función al valor de la ocupación que el conjunto de datos nos aporta. Estos colores son tres. El color verde representa una ocupación baja de la carretera, el naranja una ocupación media y el rojo hace referencia a una zona con bastante tráfico. También se le ha añadido una leyenda al final del mapa y un título indicando lo que representa:



Figura 79. Imágenes pantalla que representa la ocupación de tráfico en Madrid en tiempo real

- **Implementar las visualizaciones:** Este apartado se ha realizado al completo en la herramienta de visualización utilizada. En ella se ha usado una nueva versión que permite diseñar las gráficas generadas tanto para ordenador como para dispositivos móviles. No obstante, el principal problema se ha tenido en la implementación en la aplicación, debido a los diferentes paquetes que Power BI proporciona. La solución utilizada finalmente, como se ha comentado en la parte de visualización, ha sido incluir un link con permiso para todos los usuarios de la Universidad Europea que, para ver los cuadros de mando, tendrán que iniciar sesión. En la aplicación hemos generado la siguiente pantalla:



Figura 80. Imágenes pantalla que nos lleva a las visualizaciones creadas para el análisis de datos en la aplicación

Tras acceder al botón señalado e indicar que quieres abrir los datos usando la aplicación propia de Power BI y, teniendo en cuenta las limitaciones de inicio de sesión que se tienen actualmente, se llega a una versión propia para dispositivos móviles implementada en la propia herramienta de visualización y que tiene el siguiente aspecto:



Figura 81. imagen índice de las visualizaciones creadas para el análisis de datos desde la aplicación de Power BI para dispositivos telefónicos

- **Creación de un modelo predictivo:** Esta pantalla está dedicada a dar respuesta a los usuarios sobre la hora idónea en la que circular por un lugar determinado, un día concreto de la semana y en un intervalo horario señalado por el usuario mediante un formulario que se mostrará en la aplicación. Una vez generado el formulario, se almacenan las respuestas y son con las que se filtran los datos y se realizan los diferentes cálculos matemáticos para su respuesta.

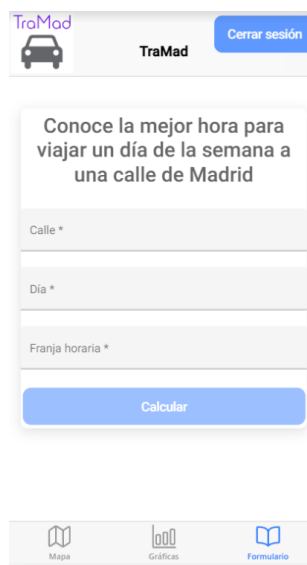


Figura 82. imagen página con el formulario para el modelo

Una vez implementada en todas las pantallas la información que se quiere reflejar, es momento de modificar el estilo de las vistas, indicando su tamaño y estilo de sus botones. Así mismo, se ha definido la ubicación de cada elemento y se han añadido tarjetas para agrupar la información. También, se ha añadido un logo y un nombre de la aplicación 'TraMad' en todas las pantallas.

Durante todo el desarrollo de la aplicación han ido apareciendo diferentes dificultades. No obstante, destacar las siguientes:

- **Nombre del fichero de los datos:** El formato en el que se almacenan los datos no coincidía con el que devolvía la aplicación, era necesario añadir un '0' delante de cada valor inferior a 10. También se tuvo que tener en cuenta que los datos se almacenaban cada 10 minutos, es decir, no hay ningún valor para las '01:46'. Tras una serie de condiciones, esto se ha conseguido solucionar.
- **Tiempo ejecución modelo:** Tras indicar todas las variables solicitadas en el modelo y enviar este, el tiempo de respuesta es bastante alto. Una posible solución para realizar a futuro es la de generar una ejecución distribuida. No obstante y, pese al tiempo, se obtiene un resultado coherente.
- **Representación de visualizaciones con Power BI:** Como se ha mencionado anteriormente, debido al paquete de la cuenta no es necesario obtener permisos de publicación de los cuadros de mandos generados para todo el público.

Una vez que se tiene toda la aplicación generada es hora de publicarla, para que los usuarios puedan acceder a ella. En este caso en concreto, la creación estaba dividida en dos carpetas diferentes, una para el back y otra para el front. De cara a su publicación se han tenido que unificar en un mismo trabajo.

Tras unificar ambas carpetas, es momento de publicar la APP para el público. Los diferentes pasos que se han realizado para el despliegue han sido los siguientes:

- **Unificación del proyecto back y front:** En este apartado ha sido necesario crear una única carpeta que englobe ambos desarrollos realizados en un primer momento por separado.
- **Crear una cuenta en Heroku:** Para el despliegue se utiliza esta plataforma, es necesario crear una cuenta y añadir una base de datos (MySQL) en la que se guarden los datos de los usuarios.

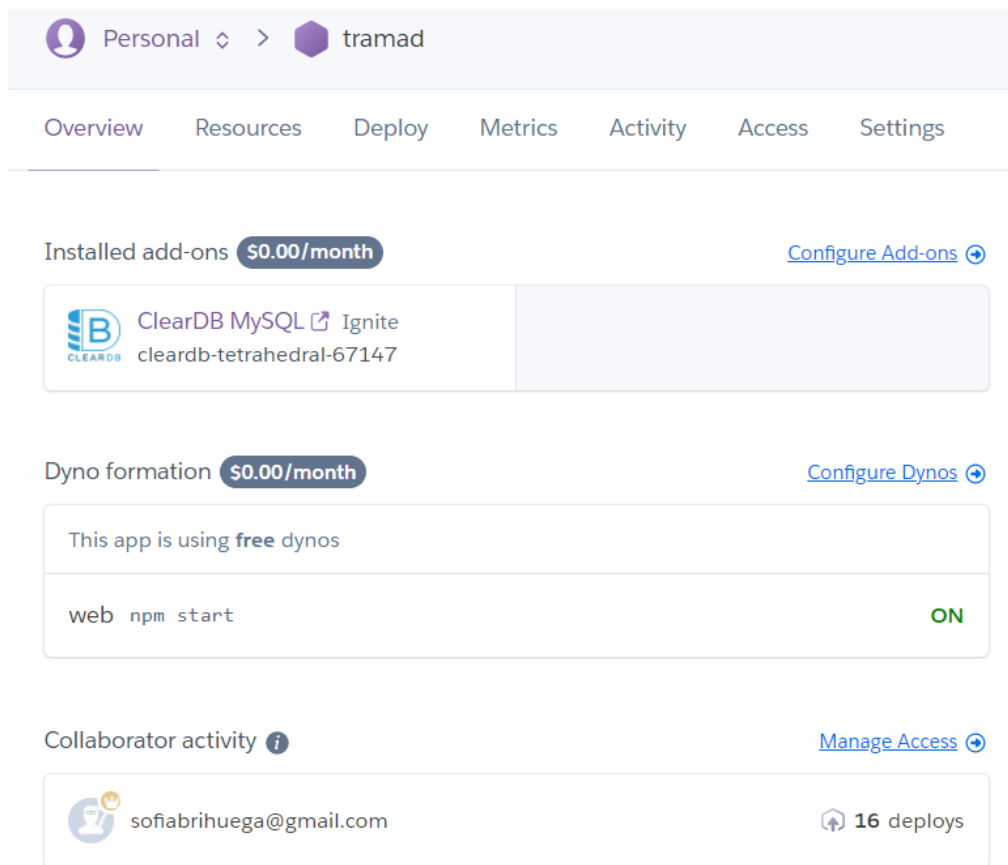


Figura 83. imagen de la cuenta cuenta y ddbb usada en Heroku

- **Construir la aplicación:** Para este apartado ha sido necesario hacer un build y subir el código de la aplicación en local a la plataforma elegida para el despliegue.

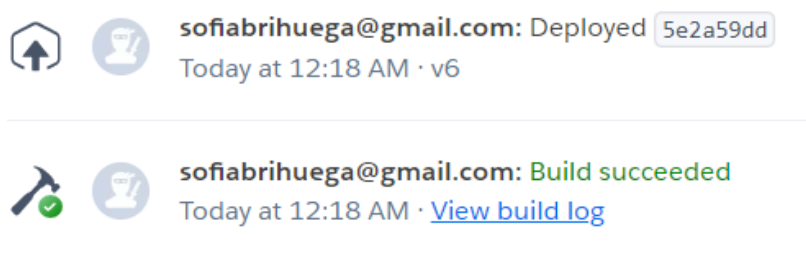


Figura 84. imagen de la construcción realizada en Heroku

- **Paso final del despliegue:** Tras subirla, el siguiente paso ha sido crear el despliegue, unificar los documentos y la base de datos que necesitaba la aplicación para funcionar. Con este último apartado se ha obtenido un URL para acceder a la aplicación, el cual se ha convertido en un código QR.



Figura 85. Código QR que nos lleva a la pantalla de iniciar sesión de la aplicación creada como producto final

Durante todo este proceso de despliegue el mayor problema que se ha encontrado ha sido que el proyecto pesaba demasiado e indicaba un error al pusearlo en Heroku, finalmente se consiguió reducir el tamaño y se pudo subir sin problema.

Creación de una marca y logo:

Una vez que se han llevado a cabo todas las actividades relacionadas con el desarrollo del trabajo fin de grado, se vio la necesidad de darle nombre a la aplicación y logo, es decir, crear una Marca. De ahí surgió la idea de **TraMad** un nombre corto, sencillo de comunicar y recordar, y que engloba en una palabra el significado de la aplicación 'tráfico en Madrid'. En cuanto al logo, se quiso crear uno sencillo con el nombre de la aplicación y una imagen que lo definiera como es el caso de un turismo De ahí surgió el siguiente logo:



Figura 86. Logo de la aplicación y del proyecto 'TraMad'

4.2.2 Metodología

Durante el desarrollo del proyecto se ha llevado a cabo una metodología Scrum compuesta por diferentes sprint, cada uno de ellos con una duración aproximada de 3 semanas. Al finalizar cada sprint se presentó al tutor los pequeños resultados obtenidos en relación a las tareas señaladas para desarrollar. Los diferentes hitos o resultados finales que se han presentado al tutor en la reunión planificada con motivo del fin del sprint son los siguientes:

Sprint 1	Documento análisis del proyecto previo y planificación
Sprint 2	Fichero de ejemplo del conjunto de datos recopilado en local

Sprint 3	Mejoras realizadas sobre el conjunto de datos
	Documento con el análisis de los datos obtenidos en local
	Prototipo o idea de la aplicación como producto final
Sprint 4	Fichero de ejemplo recopilado en GCP
	Pantallas de la APP y su interacción
	Formulario del modelo
Sprint 5	Primeros resultados del modelo
	Pantalla registro y login en la APP
Sprint 6	Visualizaciones del análisis de datos y conclusiones
	Mejores resultados en el modelo de datos
Sprint 7	Implementación del mapa en la APP
	Implementación de más visualizaciones en la APP
Sprint 8	Entrega de la memoria final
	Visualización de la APP final

Tabla 5. Resumen de los diferentes sprints llevados a cabo en el proyecto usando metodología scrum

Además de las revisiones con el tutor al finalizar el sprint, cada semana se mantenía una breve reunión en la cual se comentaban los pequeños avances obtenidos a lo largo de la semana. También se repasaban de nuevo los futuros pasos a realizar y se resolvían las diferentes dudas que habían surgido en cuanto a la continuación del proyecto.

En caso de que surgiera alguna duda urgente o bloqueo para el avance del proyecto, también se contactó con el tutor para resolver el problema rápidamente en una nueva reunión o mediante diversos mensajes por correo electrónico.

Como reflexión final, cabe destacar que, con este formato de metodología se ha podido avanzar todas las semanas en mayor o menor medida y, al implementar una metodología ágil hemos podido ver en cada entrega cómo avanzaba el producto final, es decir, en cada sprint, a excepción de los primeros, hemos podido enseñar al tutor el producto final, la aplicación y, en función del avance de las semanas, se mostraba el progreso sobre el propio producto.

El proceso llevado a cabo durante el proyecto de una manera visual tiene el siguiente aspecto:

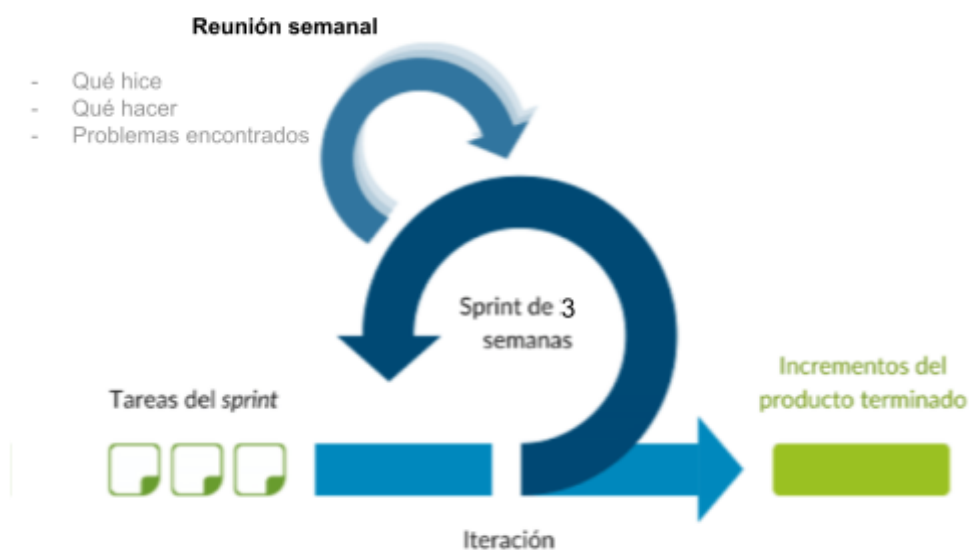


Figura 87. Esquema de la metodología llevada a cabo durante el desarrollo del proyecto

4.2.3 Herramientas empleadas

Las tecnologías empleadas durante el proyecto son las siguientes:

- Lenguajes de programación
 - Python: Principalmente enfocado en la extracción de los datos y la creación del modelo.
 - JavaScript: Usado para la aplicación o producto final.
 - CSS: Puesto en práctica en el desarrollo de la aplicación.
 - HTML: Enfocado en la aplicación.
 - JSON: Para la aplicación generada a los usuarios finales.
 - MySQL: Utilizado en el desarrollo de la base de datos de los usuarios de la aplicación.
 - node.js: Código utilizado a lo largo del desarrollo de la aplicación creada como producto final para los usuarios.

Sofía Corral Caballero

- IDEs
 - Jupyter Notebook
 - Visual Studio Code
- Distribuciones
 - Anaconda
 - GCP
 - Heroku
- Frameworks
 - Angular
 - Ionic
- Control de versiones
 - Git
- Gestor de paquetes
 - Pip
- Diseño
 - Miro
 - Draw.io
- Visualización de datos
 - Power BI
- Consultas de datos
 - MySQL Workbench
 - Postman

4.3 Recursos requeridos

Los recursos requeridos para llevar a cabo este proyecto son principalmente computacionales. Para ello, es necesario:

- **Ordenador:** con un sistema operativo actualizado de Windows. En este ordenador se ha instalado todo el Software necesario para el desarrollo del proyecto mencionado en el apartado anterior, programas y librerías. También ha sido necesaria la creación de una cuenta de Google Cloud para hacer uso de su plataforma en la nube e introducir en esta misma una tarjeta bancaria para iniciar el periodo de prueba de la plataforma.
- **Dispositivo móvil:** para testear la aplicación generada como producto final para los usuarios con acceso a internet. Las pruebas se han realizado en diferentes teléfonos móviles para verificar su funcionamiento independientemente del sistema operativo del dispositivo.

4.4 Presupuesto

Dentro del presupuesto se ha tenido en cuenta tanto las horas empleadas por el tutor como por la persona encargada de desarrollar el proyecto, incluido el material o herramientas necesarias para realizar el proyecto. En cuanto al total de horas empleadas por el personal que ha formado parte del proyecto, llegan a un total aproximado de 470 horas. En cuanto al resto de recursos necesarios, como son el ordenador, monitor o dispositivos telefónicos para el desarrollo del proyecto, no han supuesto ningún coste. No obstante, si se hiciera el

presupuesto sin tener en cuenta los recursos ya adquiridos, sería necesaria una inversión de 1.168,81€. Valor que puede aumentar si no tenemos en cuenta que se han iniciado sesiones en diversas herramientas como pruebas gratis y, por consiguiente, estas no tienen ningún coste actualmente. No obstante, se puede ver la información más detallada en la siguiente tabla:

Tipo de coste	Valor	Comentarios
Horas de trabajo en el proyecto		
Horas personales de trabajo	450 horas	Duración estimada del proyecto dividido en un total de 4 meses y medio.
Horas del tutor	20 horas	Duración estimada del total de horas invertido en reuniones y resolución de dudas.
Equipo técnico utilizado		
Ordenador portatil - Huawei MateBook D	599,00 €	Recurso ya adquirido, precio actual en el mercado.
Monitor LG	189,00 €	Recurso ya adquirido, precio actual en el mercado.
Dispositivo telefónico - Xiaomi Redmi Note 9 PRO	257,81 €	Recurso ya adquirido, precio actual en el mercado.
Dispositivo telefónico - Iphone 7	123,00 €	Recurso ya adquirido, precio actual en el mercado.
Servidor en la nube - GCP	0 €	Se ha utilizado la prueba gratuita.
Heroku	0 €	Se ha utilizado la prueba gratuita.
Software utilizado		
Jupyter Notebook	0 €	No requiere pago por licencia
Visual Studio Code	0 €	No requiere pago por licencia
Postman	0 €	No requiere pago por licencia
MySQL Workbench	0 €	No requiere pago por licencia
Power BI	0 €	No requiere pago por licencia
Librerías	0 €	No requiere pago por licencia
Estudios e informes		
Estudios e informes	0 €	Todos los informes de consulta o revistas de investigación utilizados a lo largo del proyecto eran gratuitos.
Materiales empleados		

Materiales empleados	0 €	No se ha utilizado ningún material adicional para el desarrollo del proyecto.
----------------------	-----	---

Tabla 6. Presupuesto total del proyecto desglosado

4.5 Viabilidad

Teniendo en cuenta el coste del producto generado, se puede afirmar que la viabilidad del proyecto es positiva gracias a la implementación de una aplicación como producto final. Hasta el momento, los únicos gastos del proyecto han sido los del personal, ya que el material empleado estaba adquirido previamente y las herramientas utilizadas para el desarrollo del proyecto no requerían ningún coste por el momento, al ser la mayoría gratis o al poder usar una versión de prueba gratuita.

La manera de poder generar valor con dicha aplicación se puede ver enfocada en diferentes partes:

- **Anuncios:** Una manera muy utilizada para conseguir amortizar el dinero invertido en la creación de la aplicación es mediante la incorporación de anuncios publicitarios relacionados con el tema del producto.
- **Datos:** Al entrar a la aplicación es necesario un registro donde se le solicitará al cliente datos personales como el teléfono, email, nombre y contraseña. Estos datos son almacenados en una base de datos con la finalidad de poder obtener valor con ellos. Además, es importante mencionar que las variables solicitadas son solo un ejemplo, pero se puede obtener más información relevante como el tipo de coche que tiene o la antigüedad de este, entre otras muchas opciones.

No obstante, no se puede olvidar que, al trabajar con datos en tiempo real, es necesario continuar haciendo uso de la plataforma de Google Cloud que, hasta el momento ha sido gratuita gracias a la versión de prueba que GCP ofrece durante un periodo de 90 días o un coste total de 300\$ aproximadamente. Una vez superado este periodo o coste de prueba, se empieza a cobrar en función al uso de la herramienta y los diferentes servicios que la componen. Este coste no será muy elevado puesto que el proceso más costoso que se lleva a cabo es el de almacenamiento de los datos y, estamos almacenando un total de 1,4 GB cada quince días aproximadamente.

Pese a este posible gasto con el uso de Google Cloud Platform, la herramienta sigue siendo viable si cada mes se siguen registrando usuarios nuevos que nos aporten datos y que, al crecer, nos ayude a encontrar más empresas que quieran anunciarse en nuestra aplicación o que los anuncios supongan un mayor coste y, por consiguiente, un mayor ingreso.

En caso de las visualizaciones, sería necesario hacer un análisis más exhaustivo de los diferentes paquetes que esta herramienta nos proporciona, su coste y los usuarios que van a poder acceder a ella. Quizá el amplio coste de esta herramienta sea uno de los mayores problemas para la viabilidad de la aplicación.

En cuanto al futuro de la aplicación, también será necesaria una gran inversión de captación de nuevos clientes por un lado y de mejoras en el producto por otro. Este último quizá es el más importante puesto que es necesario ir implementando constantemente nuevas

funcionalidades con la finalidad de fidelizar a los usuarios que ya están registrados y de captar a nuevos clientes. Debido al gran coste que supone estas inversiones, sería necesario realizar un análisis más en profundidad para ver si realmente es viable a futuro el proyecto, e intentar idear posibles nuevas implementaciones, así como la ampliación del alcance a más ubicaciones dentro de la comunidad o del país.

4.6 Resultados del proyecto

Como resultado final se ha logrado crear una aplicación en la que se ha implementado información de datos en tiempo real, gráficas de análisis de datos histórico y un modelo de regresión.

Durante el desarrollo del proyecto, la planificación planteada en un primer momento se ha tenido que modificar por varios motivos. El primer motivo de todos fue la dificultad de recopilar los datos en tiempo real. La idea principal era hacerlo en local y, tras una serie de pruebas, me di cuenta de que el camino que se había seguido no era el correcto debido a que al tener que estar ejecutando la función constantemente iba a ser necesario tener un ordenador encendido todo el tiempo y, esto era inviable. Para solucionar este contratiempo, decidimos usar GCP para extraer y almacenar los datos en la nube y, por consiguiente, el tiempo previsto para esta parte del proyecto se tuvo que ampliar.

Una vez modificada toda la planificación era mucho más difícil poder llegar al objetivo final previsto y crear un modelo muy desarrollado. Por lo tanto, se decidió crear un modelo más simple pero aún así que los datos que se mostraran fueran coherentes y tuvieran sentido.

No obstante, pese a los pequeños imprevistos surgidos durante el proyecto se han conseguido realizar todos los apartados planificados e implementarlos en la aplicación. Para poder comprobar que el resultado final es correcto, se han realizado diferentes pruebas que verifican la veracidad de la información representada mediante la aplicación, tanto a la hora de registrarte o iniciar sesión como de cara a la información que se pretende mostrar al usuarios.

Las diferentes pruebas que se han ido realizando sobre el producto final y de cara a la obtención de unos resultados correctos del proyecto son:

- **Comprobar la conexión entre las pantallas de inicio de sesión y crear cuenta:** Esto es importante para los usuarios puesto que al acceder a la aplicación, la primera pantalla que aparezca será la de iniciar sesión. En caso de que el cliente no tenga cuenta y este apartado no funcione no habrá manera de poder acceder a la aplicación. Para comprobar este apartado únicamente ha sido necesario acceder a la aplicación y comprobar la interacción entre ambas páginas.
- **Verificar que la base de datos funciona:** En este apartado se han realizado diferentes pruebas. En primer lugar, se ha intentado iniciar sesión con una cuenta no existente todavía en la aplicación, posteriormente se ha creado una cuenta y probado a iniciar sesión con ella. A su vez, se ha comprobado que los datos se han guardado en la base de datos de usuarios y que existen restricciones para rellenar los datos en el formulario, es decir, que el correo debe contener un '@' y un '.' o que el número de teléfono será un número.

- **Modificación del mapa en función a la hora:** En esta prueba ha sido necesario verificar en diferentes franjas horarias que los datos que representa el mapa varían en función a la ocupación que hay en esa hora en concreto.
- **Comprobar que los alumnos de la universidad Europea pueden acceder a las visualizaciones:** Para este apartado ha sido necesaria la ayuda de otros alumnos de la propia universidad para, verificar si podían ver los diferentes cuadros de mandos creados sin ningún problema.
- **Verificar el diseño responsive que Power BI nos ofrece:** Para este apartado, se han realizado diferentes pruebas accediendo desde diferentes dispositivos móviles y cuentas mediante la propia aplicación o la web. La verificación de este apartado se ha visto reflejada en diferentes imágenes del documento.
- **Comprobar el formulario de la página del modelo:** En este apartado se ha probado a rellenar el modelo de diferentes maneras, la rapidez de buscar una calle en concreto y hasta que todos los campos obligatorios no estén rellenos, no se active el botón para mandar el formulario.
- **Comprobar los resultados del modelo:** En este caso se han realizado diferentes pruebas con variables diferentes para verificar su funcionamiento. Sobre todo se ha tenido en cuenta que al filtrar por una franja horaria, el resultado que nos dé se encuentre en dicho intervalo seleccionado. Algunas de las pruebas realizadas directamente sobre la aplicación son las siguientes:
 - **Rellenar el formulario de manera aleatoria:** En este apartado se comprobará que funciona.
 - **Modificar la franja horaria:** En esta prueba se verificará que el modelo varía en función a la franja horaria.
 - **Modificar el día de la semana:** Dejar ambas variables (nombre de la calle y franja horaria) igual que en el anterior apartado y modificar el día de la semana para verificar que varía.
 - **Modificar la calle:** De la misma manera que en los apartados anteriores, se pretende verificar que el modelo varía en función a las tres aplicaciones solicitadas en el formulario.

Todas estas pruebas se han realizado en el ordenador y verificado aquellas que ha sido posible en diferentes dispositivos móviles con diferente sistema operativo, ya que tras desplegar la aplicación y esta pesar mucho, el modelo y la visualización del mapa en tiempo real no funcionan en todo momento. La manera de solucionar dicho problema es ampliando el plan gratuito que Heroku ofrece.

Capítulo 5. DISCUSIÓN

Este proyecto comenzó con la idea de ayudar a los usuarios de la aplicación a disminuir el uso de energía al volante. Para ello se planteó enfocar la solución en la velocidad a la que se circulara, pero debido al conjunto de datos y al hacerlo sobre el municipio de Madrid, se tuvo que cambiar este formato y enfocarlo más a ayudar a predecir la hora idónea a la que pasar por un punto en concreto teniendo en cuenta en este caso la variable ocupación de la carretera. No obstante, la finalidad continúa siendo la misma pese a ofrecer una solución diferente y el producto final también.

Para el desarrollo del proyecto se pretendía coger los datos en tiempo real y aquí es donde se tuvieron que realizar diferentes cambios en la planificación del proyecto debido a que duró más de lo previsto por la necesidad de tener que cambiar de herramienta.

Con la ampliación de la tarea de extracción de los datos y, pese al ajustado tiempo, se tuvo que disminuir el plazo de realización de otras tareas. En particular, del modelo del proyecto, ya que en un primer momento se pretendía tener en cuenta los datos de la ruta y de todos aquellos sensores por los que pasará en el trayecto. Este planteamiento estaba enfocado a ir entrenando el modelo por partes siguiendo con la metodología llevada a cabo en el proyecto. Primero se realizaría con el destino, posteriormente con el origen y destino y finalmente ir añadiendo datos de sensores colocados en el trayecto. Con la limitación de tiempo únicamente se ha podido implementar el modelo con los datos de destino. No obstante, se ha creado también un modelo de series temporales haciendo uso de redes neuronales que, por la falta de tiempo como se ha mencionado anteriormente y la limitación de trabajar en local, no se ha podido sacar mejores resultados.

Los datos son muy importantes para que todos los resultados implementados en la aplicación sean correctos. Por ello, se ha decidido invertir más tiempo en su extracción, análisis y modificación. Además, se ha obtenido un producto final con el mismo número de apartados que en la primera planificación.

La metodología planteada en un primer momento ha sido la que se ha llevado a cabo a posteriori, siendo esta una buena decisión pues ha sido necesario el trabajo semanal durante toda la duración del proyecto, distribuyendo así la carga de trabajo, realizando actividades en paralelo y pudiendo ver el producto final desde prácticamente los primeros sprints.

Capítulo 6. CONCLUSIONES

6.1 Conclusiones del trabajo

Con el aumento del precio de los carburantes y la electricidad, la gente ha comenzado a intentar ahorrar más en la carretera y buscar ayudas para conseguirlo. Con el desarrollo del proyecto se ha conseguido crear una aplicación basada, por un lado, en aportar información en tiempo real, añadiendo visualizaciones más genéricas enfocadas al análisis a largo plazo de una zona y, por otro lado, basada en la creación de un modelo de predicción.

El objetivo principal que se pretende llevar a cabo para los usuarios de TraMad, es disminuir el número de horas al volante en un momento en el que el tiempo personal es cada vez más valioso y la gente aprecia más tener tiempo libre para ellos, para poder disfrutar junto a sus hijos, hacer deporte o descansar entre muchas opciones. También, se pretende reducir el consumo de carburante o electricidad y, por consiguiente, de accidentes llegando a hacer más positivo el trayecto diario de los clientes de la aplicación.

Durante la elaboración del proyecto han surgido diferentes imprevistos, pero todos ellos han sido solventados con éxito llegando a obtener un resultado final muy similar al establecido en el planteamiento del proyecto.

Es importante destacar el amplio trabajo empleado en la extracción de los datos y en su posterior análisis, un trabajo muy importante de cara a obtener mejores resultados en el resto de tareas.

6.2 Conclusiones personales

Realizar este proyecto ha sido un reto satisfactorio pues era un trabajo grande y que aborda muchos puntos diferentes vistos a lo largo de la carrera. El resultado obtenido es muy positivo pues se he sido capaz de seguir en mayor medida la planificación planteada y sacar adelante las diferentes actividades propuestas.

Hasta el momento, en los trabajos realizados, siempre grupales, aunque siempre he estado presente en las diferentes partes en las que se puede dividir un proyecto, es inevitable centrarse más en una de ellas. Cuando uno solo se dispone frente al ordenador a pensar una idea, posibles soluciones, crear el código, solucionar errores, afrontar nuevos retos o crear una documentación al completo, empiezan a surgir dudas, y es cuando uno se da cuenta de la verdadera complejidad del trabajo. Superar todos los problemas, podríamos decir que es el reto más difícil de todo este trabajo y a su vez la carrera al completo.

Haber conseguido implementar todo lo previsto y la mayoría de los conceptos aprendidos durante la carrera es muy gratificante.

Durante gran parte de la carrera el coche ha sido el medio de transporte personal diario a la universidad. En cada viaje que hacía se ha intentado plantear diferentes soluciones para poder evitar los atascos que se sufrían día tras día, intentando hacer uso de algún concepto aprendido nuevo en clase. El TFG es el trabajo más importante de la carrera e intentar realizarlo de algo que me ha acompañado durante estos cuatro años es algo que personalmente me parece una buena idea.

Capítulo 7. FUTURAS LÍNEAS DE TRABAJO

Existen diferentes apartados en los que desarrollar un poco más el proyecto realizado pero que principalmente por tiempo no se han podido implementar en esta primera entrega. Algunas ideas son:

- **Ampliar los datos:** En la actualidad, únicamente se está recogiendo información del municipio de Madrid en tiempo real. Sería muy positivo ampliar el conjunto de datos en un primer lugar a toda la Comunidad, posteriormente al país entero y finalmente añadiendo otros países.
- **Realimentación del modelo:** Actualmente, los resultados de ambos modelos se realizan entrenando unos datos fijos, es decir, se tiene un dataset con la información recogida del municipio de Madrid a lo largo de varias semanas y de ahí es de donde se sacan los resultados del modelo. Para que este de mejores resultados, sería muy recomendable ir aumentando dicho dataset con los datos que se van almacenando cada diez minutos en el bucket creado en Google Cloud Platform.
- **Seguir desarrollando el modelo:** Hasta ahora, el modelo nos dice en función a la calle y el día de la semana que se quiere viajar, la hora en la que hay menos tráfico. A este modelo se le puede ir incrementando nuevas funciones como por ejemplo que en vez de decirle el día de la semana se le diga el lugar y la hora a la que se quiere pasar por ahí y él nos diga el día de la semana en el que habrá menos tráfico. Este modelo puede ser muy interesante en la actualidad para aquellas personas que tienen que ir a la oficina únicamente un día por semana, al implementar este nuevo modelo le podemos ayudar a decidir qué día le conviene más viajar. También, se puede seguir desarrollando el modelo añadiendo una ruta total y no solo teniendo en cuenta el punto de destino, es decir, ver por todos los puntos por los que se pasa en el trayecto y en el que menos tráfico se va a encontrar.
- **Visualizaciones de Power BI:** Los diferentes cuadros de mandos desarrollados como análisis de datos hasta el momento no los puede ver todo el mundo, esto es algo que limita a los usuarios de la aplicación a acceder. Como trabajo a futuro se debe intentar conseguir hacer públicas las vistas generadas en Power BI.
- **Velocidad en el modelo:** Con los recursos que se utilizan hasta el momento, el modelo tarda un largo tiempo en dar respuesta al usuario, como trabajo a realizar para mejorar el producto se podría introducir ejecución distribuida y hacer que disminuya el tiempo de carga del resultado.

Capítulo 8. REFERENCIAS

- [1] Uso, P. (2020, 6 julio). El estrés durante un atasco aumenta el riesgo de accidente “in itinere”. Sindicato USO.
<https://www.uso.es/el-estres-durante-ataasco-aumenta-riesgo-de-accidente-in-itinere/>
- [2] Pérez, L. (2017, 17 abril). ¿Por qué se forman los atascos? Motor y Movilidad.
<https://www.20minutos.es/noticia/2839009/0/porque-se-producen-los-atascos/>
- [3] Accidentes de tráfico, en datos y estadísticas. (2022). EpData.
<https://www.epdata.es/datos/accidentes-trafico-datos-estadisticas/65/espana/106>
- [4] elEconomista.es. (2021, 28 junio). Así es el «efecto acordeón», cómo se evita uno de los accidentes más repetidos y qué multas conlleva.
<https://www.eleconomista.es/actualidad/noticias/11295735/06/21/Asi-es-el-efecto-acordeon-como-se-evita-uno-de-los-accidentes-mas-repetidos-y-que-multas-conlleva.html>
- [5] Accidentes de tráfico, en datos y estadísticas. (2022b, enero 7). epdata.
<https://www.epdata.es/datos/accidentes-trafico-datos-estadisticas/65/espana/106>
- [6] Casado, D. (2021, 5 octubre). Madrid vuelve a los niveles de atascos previos a la pandemia: «Es decepcionante no haber aprendido nada». ElDiario.es.
https://www.eldiario.es/madrid/somos/noticias/madrid-vuelve-niveles-atascos-previos-pandemia_1_8363832.html
- [7] Informes de tráfico en hora punta - Ayuntamiento de Madrid. (s. f.). Madrid.es.
<https://www.madrid.es/portales/munimadrid/es/Inicio/Movilidad-y-transportes/Informes-de-trafico/Informes-de-trafico-en-hora-punta/?vgnnextfmt=default&vgnextoid=9090391591232710VgnVCM2000001f4a900aRCRD&vgnnextchannel=71018e88b5632710VgnVCM2000001f4a900aRCRD>
- [8] Grasso, D., & Andrino, B. (2021, 12 diciembre). ¿Cómo ha cambiado la movilidad en Madrid? Un millón de viajes menos en transporte y coches como antes. El País.
<https://elpais.com/espana/madrid/2021-12-12/como-ha-cambiado-la-movilidad-en-madrid-un-millon-de-viajes-menos-al-dia-en-transporte-y-trafico-como-antes-de-la-pandemia.html>
- [9] EMT Madrid. Empresa Municipal de Transportes de Madrid, S. A. - Plan estratégico. (s. f.). EMTMadrid.
<https://www.emtmadrid.es/Elementos-Cabecera/Enlaces-Pie-vertical/EMPRESA/Somos/Plan-Estrategico>
- [10] EMT Madrid. Empresa Municipal de Transportes de Madrid, S. A. - Plan estratégico. (s. f.). EMTMadrid.
<https://www.emtmadrid.es/Elementos-Cabecera/Enlaces-Pie-vertical/EMPRESA/Somos/Plan-Estrategico>
- [11] ‘Muévete en EMT’, la campaña para fomentar el transporte público en Madrid - Ayuntamiento de Madrid. (2020, 17 noviembre). Madrid.es.
<https://www.madrid.es/portales/munimadrid/es/Inicio/Actualidad/Noticias/Muevete-en-EMT->

la-campana-para-fomentar-el-transporte-publico-en-Madrid/?vgnnextfmt=default&vgnnextoid=2d2a4957295d5710VgnVCM1000001d4a900aRCRD&vgnnextchannel=a12149fa40ec9410VgnVCM100000171f5a0aRCRD

[12] Matriculaciones de turismos en España. (2022, 1 abril). EpData.

<https://www.epdata.es/datos/matriculaciones-turimos-espana-carburante/206/espana/106>

[13] Precio de la factura de la luz, datos y estadísticas. (2022, 27 marzo). EpData.

<https://www.epdata.es/datos/precio-factura-luz-datos-estadisticas/594>

[14] Precio de la gasolina y gasóleo (diesel) en España, cifras, datos y estadísticas. (2022, 31 marzo). EpData.

<https://www.epdata.es/datos/precio-gasolina-gasoleo-diesel-espana-cifras-datos-estadisticas/420>

[15] Pérez, C. (2021, 31 marzo). Qué es Waze, para qué es útil y cómo funciona.

MuyInteresante.es.

<https://www.muyinteresante.es/tecnologia/articulo/que-es-waze-para-que-es-util-y-como-funciona-741617145169>

[16] Veliz, J. (2020, 29 junio). Waze quiere tu atención para su nueva versión: cambia de logo, de colores y de ánimo. RPP.

<https://rpp.pe/tecnologia/apps/waze-este-es-el-rediseño-que-ha-recibido-la-aplicación-como-usar-los-nuevos-moods-para-el-trafíco-apps-gps-noticia-1276487>

[17] Tráfico. Datos del tráfico en tiempo real - Portal de datos abiertos del Ayuntamiento de Madrid. (s. f.). datos Madrid.

<https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbc4b2e4b284f1a5a0/?vgnnextoid=02f2c23866b93410VgnVCM1000000b205a0aRCRD&vgnnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD>

Enlaces genéricos necesarios para el desarrollo

Google Cloud Platform

- Saenz, M. E. (2019, 1 agosto). Cómo programar un script de Python en GCP. CloudEspañol.
<https://www.xn--cloudespaol-9db.com/2019/08/como-programar-un-script-de-python-en.html>
- Google Cloud Free Program. (s. f.). Google Cloud.
<https://cloud.google.com/free/docs/gcp-free-tier/#workflows>
- Free Trial and Free Tier |. (s. f.). Google Cloud.
https://cloud.google.com/free/?utm_source=google&utm_medium=cpc&utm_campaign=emea-none-all-none-dr-sitelink-all-all-trial-e-gcp-1011340&utm_content=text-ad-none-any-DEV_c-CRE_526889180785-ADGP_Hybrid%20%7C%20BKWS%20-%20EXA%20%7C%20Ttxt%20%7E%20GCP%20%7E%20General%23v2-KWID_43700060384861660-kwd-87853815-userloc_9061033&utm_term=KW_gcp-ST_gcp-NET_g-&gclid=CjwKCAjwl

caRBhBYEiwAK341jXXTY0qJ5jKlbnqCaBdWIndDUXxOVpP7AR06FUzv4PjraiTCr_dblxoCOooQAvD_BwE&gclid=aw.ds

- Soluciones de analíticas en tiempo real | . (s. f.). Google Cloud.
<https://cloud.google.com/solutions/stream-analytics?hl=es>
- M. (2020, 24 julio). Cómo ejecutar GCP Cloud Functions periódicamente con Cloud Scheduler. WebSetNet.
<https://websetnet.net/es/how-to-run-gcp-cloud-functions-periodically-with-cloud-scheduler/>
- Jauregui, A. F. (2021, 14 febrero). Cómo automatizar un script de Python en Google Cloud. Ander Fernández.
<https://anderfernandez.com/blog/automatizar-script-python-google-cloud/>
- google-cloud-storage — Cómo descargar varios archivos en Google Cloud Storage. (2016, 25 agosto). Desarrollo-web.
<https://www.desarrollo-web-br-bd.com/es/google-cloud-storage/como-descargar-varios-archivos-en-google-cloud-storage/826929937/>
- Velasco, R. (2022, 24 enero). ¿No tienes gestor de descargas? Así puedes descargar con PowerShell. SoftZone.
<https://www.softzone.es/windows/como-se-hace/descargar-archivos-powershell/>
- Make data public | Cloud Storage | . (s. f.). Google Cloud.
<https://cloud.google.com/storage/docs/access-control/making-data-public>

Visualización

- E. (2020a, abril 3). Dashboard de medición de la calidad del aire en Madrid. TodoBI - Business Intelligence, Big Data, ML y AI.
<https://todobi.com/dashboard-de-medicion-de-la-calidad-del-aire-en-madrid/>
- M. (2022, 14 mayo). Cambiar el modo en que se ordena un gráfico en un informe - Power BI. Microsoft Docs.
<https://docs.microsoft.com/es-es/power-bi/consumer/end-user-change-sort>
- Power BI Sign in. (s. f.). Power BI.
<https://app.powerbi.com/groups/me/reports/0557e5f5-63c4-424c-9916-4fd2014c7d19/ReportSection4c215b7472ce3d1262b0>
- M. (2022a, mayo 9). Insertar un elemento web de informes en SharePoint Online - Power BI. Microsoft Docs.
<https://docs.microsoft.com/es-es/power-bi/collaborate-share/service-embed-report-spo>

Aplicación

- (2018, 15 mayo). 50 login y formularios de registro para inspirarse - UXlatam. Medium.
<https://medium.com/uxlatam/50-login-y-formularios-de-registro-para-inspirarse-96f9282e4c3e>
- Raveh, N. (2021, 19 septiembre). Announcing Power BI Angular Component. Blog de Microsoft Power BI | Microsoft Power BI.
<https://powerbi.microsoft.com/es-es/blog/announcing-power-bi-angular-component/>

- I. (s. f.). Cross-Platform Mobile App Development: Ionic Framework. <https://ionicframework.com/>
- Importar y Exportar un Base de datos en MySQL Workbench. (2019, 18 diciembre). YouTube. <https://www.youtube.com/watch?v=jTs1nSwAcRM>

Modelo

- Waas, N. (2022, 6 enero). Time Series Forecasting: Predicting mobile traffic with LSTM. Medium. <https://medium.com/analytics-vidhya/time-series-forecasting-predicting-mobile-traffic-with-lstm-649e6ba7ee49>
- Heras, J. M. (2020, 10 octubre). Error Cuadrático Medio para Regresión. IArtificial.net. <https://www.iartificial.net/error-cuadratico-medio-para-regresion/>

Capítulo 9. ANEXOS

Anexo 1. Código fuente

El código generado durante el desarrollo del proyecto está disponible en un repositorio público de GitHub al que puedes acceder mediante el siguiente enlace: <https://github.com/sofiaCorral/TraMad-TFG>. En él se pueden encontrar diferentes carpetas que dividen todo el desarrollo empleado en bloques dependiendo de su funcionamiento. Estos bloques son:

- **Database:** Se adjunta un fichero SQL con la base de datos generada para el registro de los usuarios en la aplicación.
- **Modelo:** Aparece el modelo de redes neuronales implementado como valor añadido en python.
- **Pre-procesamiento:** En este apartado se sitúan adjuntas todas las funciones generadas en GCP para la unión del conjunto de datos de cara a la realización de un análisis de datos, su implementación en la aplicación, modelo y visualizaciones posteriormente.
- **Scraping:** Se adjunta el archivo de la función creada como prueba en local y el código final ejecutado en Google Cloud Platform.
- **Visualización:** Aparece el archivo generado en Power BI con todas las gráficas, KPIs y filtros llevados a cabo durante el proyecto.
- **Aplicación:** Este apartado está compuesto por todo el código desarrollado para la aplicación.
- **Documentación:** Se adjunta la memoria generada explicando el trabajo llevado a cabo y el documento de originalidad.

