Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**

Máster en Ciencia de Datos

Trabajo Fin de Máster

Análisis Topológico de Datos aplicado a Trayectorias de Aeronaves

Autor: Raúl Serrano Campillo

Madrid, julio, 2025

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid.

Trabajo Fin de Máster
Máster en ***Ciencia de Datos***

Título: **Análisis Topológico de Datos Aplicado a Trayectorias de Aeronaves**
Julio, 2025

Autor: **Raúl Serrano Campillo**

Tutor

Alfonso Mateos Caballero

ETSI Informáticos
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid

Co-Tutor

Arminda Moreno Díaz

ETSI Informáticos
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid

Resumen

El presente Trabajo de Fin de Máster aborda el análisis de trayectorias de aeronaves mediante técnicas de Análisis Topológico de datos sobre un conjunto de datos que contiene las posiciones reales y planificadas de los vuelos operados en los aeropuertos españoles durante la temporada de verano de 2018. Este tipo de análisis surge como una herramienta innovadora y eficaz para analizar conjuntos de datos de alta dimensión, tratando de enfocarse en aquellas propiedades invariantes del conjunto de datos.

El proyecto se centra en la compleja depuración de una base de datos para finalmente aplicar el algoritmo Mapper. A partir de las trayectorias reales y planificadas basadas en coordenadas de cuatro dimensiones (latitud, longitud, altitud y tiempo) de los vuelos se pueden extraer características como son el retraso y la desviación que han tenido los vuelos. Tras la aplicación del algoritmo Mapper sobre estas características podemos identificar distintas agrupaciones de los aeropuertos de España a partir de las características de sus vuelos.

Los resultados de este análisis revelan la capacidad del algoritmo Mapper de detectar grupos de aeropuertos con comportamientos similares, que no siempre están alineados con clasificaciones tradicionales basadas en el volumen de pasajeros. Estos patrones pueden ser de gran utilidad para mejorar las estrategias de gestión del tráfico aéreo en la red aérea española.

Este trabajo demuestra el valor del Análisis Topológico de Datos para analizar las trayectorias de aeronaves, datos que suelen enmarcarse en contextos complejos y de alta dimensión. Además, se expone un análisis metodológico detallado y se abren las puertas a futuras aplicaciones y nuevas investigaciones que se apoyen en este trabajo.

Abstract

This Master's Thesis addresses the analysis of aircraft trajectories using Topological Data Analysis (TDA) techniques on a dataset containing the actual and planned positions of flights operated at Spanish airports during the summer season of 2018. This type of analysis emerges as an innovative and effective tool for examining high-dimensional datasets, focusing on the invariant properties within the data.

The project centres on the complex preprocessing of a database to ultimately apply the Mapper algorithm. Based on the actual and planned flight trajectories described by four-dimensional coordinates (latitude, longitude, altitude, and time), it is possible to extract features such as delays and deviations experienced by the flights. After applying the Mapper algorithm to these features, we can identify different groupings of Spanish airports based on the characteristics of their flights.

The results of this analysis reveal the capability of the Mapper algorithm to detect clusters of airports with similar behaviours, which do not always align with traditional classifications based on passenger volume. These patterns can be highly valuable for improving air traffic management strategies within the Spanish air network.

This work demonstrates the value of Topological Data Analysis for studying aircraft trajectories, which are typically embedded in complex and high-dimensional contexts. Furthermore, it presents a detailed methodological analysis and opens the door to future applications and new research built upon this study.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido, de una u otra manera, a la realización de este Trabajo de Fin de Máster.

En primer lugar, quiero agradecer a mi familia por todo el apoyo incondicional que he sentido. Especialmente a mi padre y a mi madre, los que han sido mis referentes, mis ídolos y han supuesto mi mayor apoyo, incluso en los momentos de mayor intensidad.

En segundo lugar, quiero agradecer a mis tutores por su guía, paciencia y dedicación durante todo el desarrollo del proyecto. Su orientación y su confianza han sido claves para superar los desafíos encontrados en este camino.

Gracias Alfonso, por tus consejos, tu sabiduría y tu compromiso constante en todo el proyecto. Tu apoyo técnico, implicación y experiencia en el tema han sido determinantes durante todos estos meses.

Gracias Arminda, por tu dedicación, tus ganas y tu ilusión tanto por enseñar como por aprender. Tu cercanía, tu constante disponibilidad o tu visión crítica han permitido aportar distintos enfoques y enriquecer enormemente este proyecto.

También quiero dar las gracias a mis compañeros y amigos del máster, con quienes he compartido ideas, dudas y motivación en los momentos de mayor intensidad académica.

This research is part of the R&D&I projects PID2021-122209OB-C31 funded by MCIU/AEI/10.13039/501100011033.

Tabla de contenido

Índice de figuras

Índice de tablas

1 Introducción

1.1 Contexto

El transporte aéreo se ha consolidado como un pilar fundamental para conseguir un mundo más globalizado y conectado. La infraestructura logística, tecnológica y operativa tiene una complejidad enorme debido a la independencia de cada sistema, así como a la cantidad masiva de elementos, pasajeros y vuelos que incluye. Solo en 2024, las compañías aéreas comerciales vendieron más de 5850 millones de billetes, sobrepasando el récord anterior de 2019 que se situaba en 5720 millones. El número total de vuelos fletados por dichas compañías ascendía a 36.4 millones, un 5% por debajo de los niveles de 2019. En lugar de programar más vuelos, se observa una tendencia por parte de las compañías a programar vuelos con aviones más grandes más que aumentar la frecuencia de estos [1].

La Asociación Internacional de Transporte Aéreo (IATA) también se hizo eco de estos valores, en máximos históricos, de la demanda de billetes de avión comerciales en el año 2024 [2]. Además, la ocupación media de los vuelos programados en 2024 fue del 83.5%. Estas cifras reflejan que no solo se recuperó el tráfico aéreo después de la pandemia del COVID-19, si no que el sector ha superado los datos prepandemia puesto que el siguiente año con más pasajeros según la IATA fue 2019. Y las previsiones para el año 2025 indican que esta demanda va a continuar creciendo.

En esta compleja red de comunicaciones, las rutas, en las que se indican las coordenadas por las que pasa la aeronave, se planifican horas antes de la salida real del vuelo. Sin embargo, adaptarse a los planes de vuelo presentados a priori es un gran reto que no suele cumplirse al pie de la letra. Las razones son múltiples e incluyen restricciones en el control del tráfico aéreo o en el espacio aéreo mismo, cambios para mejorar el flujo o evitar congestión, fenómenos y condiciones meteorológicas (turbulencias, tormentas o vientos en altura), limitaciones en las aeronaves, entre otros.

El impacto de estas diferencias se refleja directamente en los costes operativos y medioambientales. Por ejemplo, un mayor tiempo de vuelo implica mayor consumo de combustible y afecta también a la capacidad del espacio aéreo, congestionando más el flujo, así como a la fiabilidad de los sistemas que intentan predecir la gestión del tráfico basado en los planes de vuelo inicialmente planificados.

Gestionar eficientemente el tráfico aéreo es un desafío muy importante para garantizar la seguridad, puntualidad, sostenibilidad y capacidad del transporte aéreo mundial. Este proyecto pretende explorar nuevos mecanismos de representación de rutas aéreas que eventualmente puedan integrarse en esa gestión.

Sin embargo, analizar las rutas aéreas es una tarea compleja debido a múltiples factores. La gran cantidad de datos que se recogen desde los distintos puntos de radares terrestres, radares secundarios, planes de vuelo, sensores meteorológicos [3] o datos recogidos por las propias aeronaves hacen que se recojan una gran cantidad de características de un sistema muy complejo que a menudo puede acabar

produciendo conjuntos de datos de muy alta dimensión, con escasa información y muy difícil de extraer [4].

Estos conjuntos de trayectorias de vuelo son difíciles de analizar debido a:

- Las múltiples variables que se utilizan como altitud, velocidad, rumbo, longitud, latitud y tiempo; influyendo cada una en la trayectoria.
- La continuidad de los datos con un gran número de medidas debido a los registros continuos y densos en el tiempo. Estos conjuntos de datos presentan las tres características típicas: volumen, variedad (no homogeneidad) y velocidad, recolectados en tiempo real.
- La gran variedad de interacciones que existen entre ellos, como el control del tráfico aéreo, las condiciones meteorológicas, el tipo de aeronave o el comportamiento del piloto que complican el aislamiento de un determinado efecto individual.
- Los cambios dinámicos, ya que las trayectorias van variando a medida que se desplaza la aeronave e interactúa con ese entorno, dificultando encontrar patrones sin métodos estadísticos avanzados.
- La alta dimensionalidad de los datos, debido a la grandísima cantidad de características recogidas, lo que dificulta analizar los datos sin reducir su dimensionalidad con técnicas como el análisis de componentes principales (PCA).

A todas estas dificultades cabe añadir la escasa disponibilidad de datos de trayectorias de vuelos, debido a la privacidad de estos y a políticas que limitan su uso por parte de la comunidad investigadora.

1.2 Objetivos

El objetivo principal de este Trabajo de Fin de Máster es realizar un análisis detallado de una base de datos que contiene la descripción de las trayectorias de los aviones comerciales que han despegado o aterrizado en los aeropuertos españoles en un determinado periodo de tiempo. Esta base de datos ha sido proporcionada por CRIDA. Aparte de abordar el reto de convertir la información desestructurada original en un conjunto de variables estructuradas aptas para su análisis, el objetivo principal contempla también explorar las posibilidades del Análisis Topológico de Datos (TDA) y en concreto, del algoritmo Mapper, para obtener representaciones, agrupaciones e interpretaciones alternativas de los datos de trayectorias de vuelos. Además, la visualización de estas agrupaciones puede, no solo compararse con las ya existentes sino arrojar nueva información que permita comprender mejor la compleja dinámica que rige el comportamiento del espacio aéreo español.

La consecución de este objetivo principal conlleva la ejecución de una serie de tareas secundarias no menos importantes, como son:

- El análisis de trabajos previos que hayan utilizado información similar sobre trayectorias de aeronaves y su interpretación.
- El preprocesamiento y estructuración de los ficheros de datos originales, incorporando la información externa necesaria para convertirlos en un

conjunto de datos con información autocontenida y apto para ser usado en un proyecto de ciencia de datos.
- La creación, a partir de la información anterior, de variables nuevas que capturen la información relevante con la que alimentar los algoritmos, no solo de TDA sino de otra naturaleza, que se apliquen sobre ellas. En este caso, esa información se obtiene a partir del cálculo de características de los vuelos como son el retraso y la desviación del vuelo.
- El estudio del algoritmo Mapper.
- El análisis de las visualizaciones obtenidas por Mapper, así como su evaluación y comprensión para detectar patrones o agrupamientos.
- Identificar, a partir de las visualizaciones, las diferencias en el tráfico aéreo entre los distintos meses o entre distintos aeropuertos.

1.3 Trabajos previos

El análisis topológico de datos (TDA) nació a principios del siglo XX, aunque engloba también trabajos matemáticos del siglo XIX [5]. Este análisis estudia las propiedades espaciales de los objetos que permanecen invariantes tras deformaciones.

Este tipo de análisis se basa principalmente en la idea de que los datos de alta dimensión forman nubes de puntos cuyas características (o forma) tienen información relevante. Algunas de estas características pueden ser componentes conexas, agujeros o huecos [6].

En la línea de investigación que se sigue en este proyecto existen algunos trabajos previos relacionados con los métodos y objetivos que se persiguen.

En [6] se aplicó la homología persistente en los grafos obtenidos de los datos de los aeropuertos de Estados Unidos consiguiendo demostrar que los Betti-1 (bucles o agujeros unidimensionales linealmente independientes) elevados identifican cuellos de botella que pueden generar demoras sistemáticas.

En 2024 se utilizaron tanto complejos de *Vietoris-Rips* como diagramas de persistencia para clasificar trayectorias ruidosas de vehículos y barcos sin etiquetas. De esta forma lograron demostrar que la forma de la trayectoria es suficiente para discriminar estilos de vuelos [7].

En [4] se analizan trayectorias de vuelos que aterrizan o que despegan desde España generando paisajes de persistencia y demostrando que estos paisajes son capaces de correlacionarse con los retrasos de las trayectorias aéreas. Además, se compara esos aeropuertos con clasificaciones clásicas como la proporcionada por AENA en 2018. Para ello, calculan tanto las desviaciones de las rutas reales con las planificadas como el retraso de los vuelos y utilizan esas características para formar los paisajes de persistencia. El proyecto que se plantea en esta memoria tiene como punto de partida este trabajo en cuanto que utiliza la misma fuente de información, aunque con cambios en las características que se utilizan, así como el algoritmo del análisis topológico utilizado. Otra diferencia fundamental se encuentra en el filtrado y procesado de los datos de las trayectorias contenidas en los ficheros originales proporcionados por CRIDA.

1.4 Estructura de la memoria

Lo que resta de este Trabajo de Fin de Máster (TFM) está organizado en los siguientes capítulos:

El segundo capítulo aborda el proceso seguido desde la recepción de los datos originales hasta obtener un conjunto de información viable con el que poder aplicar el tipo de análisis deseado. Incluye la presentación de los datos de los que parte el proyecto, así como su depuración, limpieza y, finalmente, la obtención de las variables sobre las que aplicar el Análisis Topológico.

El tercer capítulo incluye, en primer lugar, una introducción al Análisis Topológico de Datos y al algoritmo Mapper que actúa no solo como una plataforma de visualización que permite el descubrimiento de grupos en datos de alta dimensión, sino como una herramienta para estudiar la conectividad entre regiones del espacio de datos. En segundo lugar, se muestran los resultados que se obtienen tras la aplicación de dicho algoritmo a los datos que fueron preparados en el capítulo anterior.

Por último, en el último capítulo se resumen las conclusiones obtenidas. Además, se exponen algunas de las limitaciones encontradas en el desarrollo del mismo y se proponen algunas líneas futuras de investigación que pudieran ampliar y mejorar este trabajo.

2 Metodología

En este capítulo se detallan todos los pasos que se han seguido para transformar el conjunto de datos original en un conjunto con información estructurada viable para su uso. Se pueden distinguir dos fases: manipulación de los datos originales para darles una estructura que permita acceder a la información que contienen y transformación de las variables iniciales en otras que sean aptas para el uso por parte de los algoritmos de representación aplicados. Un ejemplo de la primera fase es la transformación de cada trayectoria real y planificada en vectores de cuatro coordenadas (tiempo, longitud, latitud, altitud). Un ejemplo de la segunda fase es el uso de estos vectores para el cálculo de la distancia entre las trayectorias real y planificada. La Figura 2.1 resume los pasos de la metodología implementada.

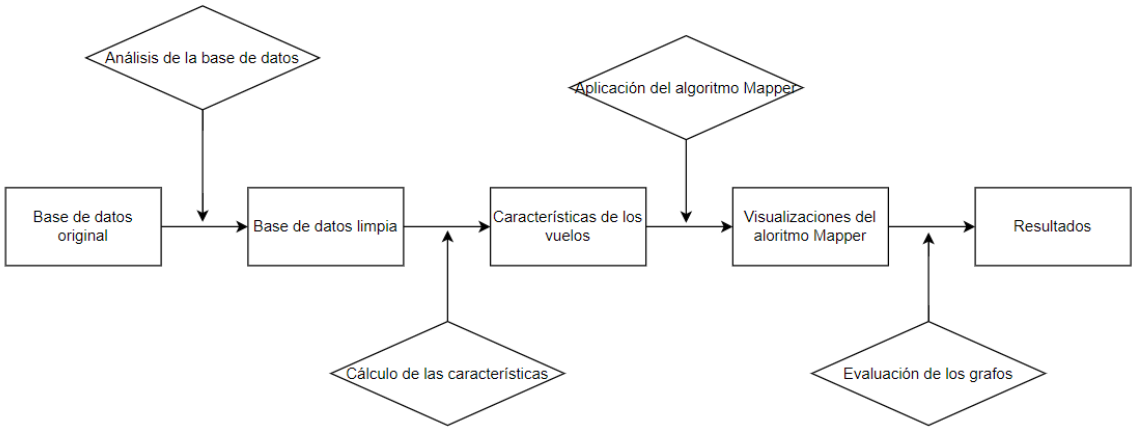

Figura 2.1 Resumen Metodología

2.1 Base de Datos Original

El conjunto de datos original ha sido proporcionado por CRIDA (Centro de referencia de investigación, desarrollo e innovación ATM, [8]) a través de la herramienta de modelado y simulación NEST [9] utilizada por la red EUROCONTROL y por los proveedores de servicios de navegación aérea (ANSP). NEST incluye en su estructura tres bloques con datos para visualizar: datos de espacio aéreo, datos de red y datos de vuelo.

Este conjunto de datos se analiza, desde otra perspectiva, en el artículo "Topological Data Analysis in Air Traffic Management: The shape of big flight data sets" [4]. Fue a través de este trabajo como tuvimos noticia de su existencia y disponibilidad.

El conjunto de datos está estructurado en 4 archivos:

- ‘readme.txt’ contiene los metadatos del resto de archivos y la descripción de los datos que contienen: además, contiene las instrucciones para abrir los archivos utilizando librerías de Python [10].
- ‘inicialesspain’ es un archivo comprimido de Python que contiene una lista de listas de los vuelos que salen o llegan a España en la temporada de verano de 2018 con las rutas planificadas. Concretamente las rutas son de los vuelos realizados entre el 25/03/2018 y el 28/10/2018.
- ‘finalesspain’ es un archivo comprimido de Python que contiene una lista de listas de los vuelos que salen o llegan a España en la temporada de verano de 2018 con las rutas reales. Concretamente las rutas son de los vuelos realizados entre el 25/03/2018 y el 28/10/2018.
- ‘meduaslandscapes_spain’ es un archivo comprimido de Python que contiene los paisajes de persistencia promedios de los aeropuertos españoles en la temporada de verano de 2018. Puesto que este archivo es un resumen del trabajo anterior, este archivo no se utiliza para este proyecto.

Los archivos con los datos que vamos a procesar son ‘inicialesspain’ y ‘finalesspain’. Estos archivos contienen los datos de los vuelos que han despegado o aterrizado en España y están estructurados de la misma forma. El primero contiene la información que representa las rutas planificadas que siguen esos vuelos. El segundo contiene la información de las rutas reales que al final han seguido esos mismos vuelos. Podemos ver la estructura de estos archivos en la Figura 2.2.

Ambos archivos son una lista en la que cada subelemento representa la información de los vuelos de un día. La información de los vuelos comprende la temporada de verano del 2018. Concretamente, los datos van desde el 25/03/2018 hasta el 28/10/2018, un total de 217 días.

Cada subelemento de la lista principal, que representa un día, es, a su vez, una lista que contiene la información de los vuelos de ese día. Por lo tanto, los archivos originales son una lista de sublistas. Juntando todos los vuelos de todos los días, la base de datos engloba un total de datos de 1.134.813 vuelos.

Cada vuelo es, a su vez, una lista de 4 elementos en la que cada uno representa la siguiente información:

- Aeropuerto de salida.
- Aeropuerto de llegada.
- Código del vuelo.
- Trayectoria del vuelo representada en un conjunto de coordenadas 4D.

Por lo tanto, para ambos archivos tendremos los mismos datos de aeropuerto de salida, aeropuerto de llegada y código del vuelo. Sin embargo, la trayectoria del vuelo será la principal diferencia entre ellos puesto que una representa la ruta planificada que debía seguir el vuelo, mientras que la otra representa la ruta real que efectivamente ha recorrido.

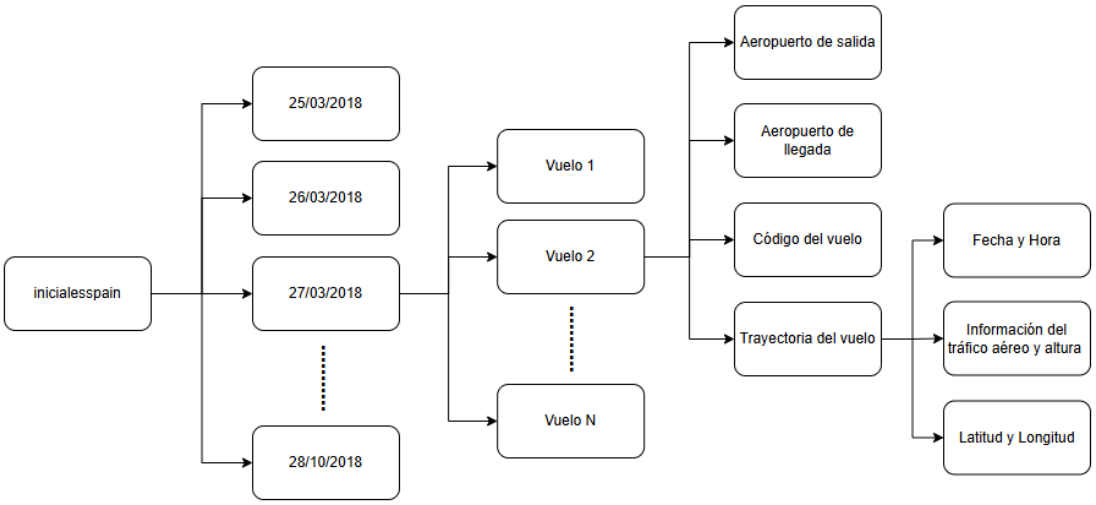

Figura 2.2 Estructura de la base de datos original

La trayectoria está estructurada como una cadena de texto que contiene cadenas separadas por un espacio en las que cada cadena representa una coordenada 4D. Por ejemplo, la siguiente cadena de caracteres:

'20180325031122:!HOrg:NO_ROUTE:90:::403705N0033227W'

se descompone en cuatro piezas de información sobre la localización del avión, concretamente:

- '20180325031122' → Representa la fecha y hora de la coordenada del vuelo.
- '!HOrg:NO_ROUTE:90' → Representa información relacionada con la gestión del tráfico aéreo. El número final representa la altura a la que se encuentra el vuelo en ese instante, medida en pies/100.
- '403705N0033227W' → Representa las coordenadas espaciales de la localización del vuelo en ese instante, en forma de latitud y longitud.

Obteniendo las coordenadas en forma de longitud y latitud de una ruta planificada y una ruta real en cada instante de tiempo para un vuelo, se puede visualizar la diferencia entre ambas. La Figura 2.3 representa en azul la ruta planificada, mientras que la línea roja representa la ruta real que ha seguido un vuelo determinado, en este caso Madrid-Moscú.

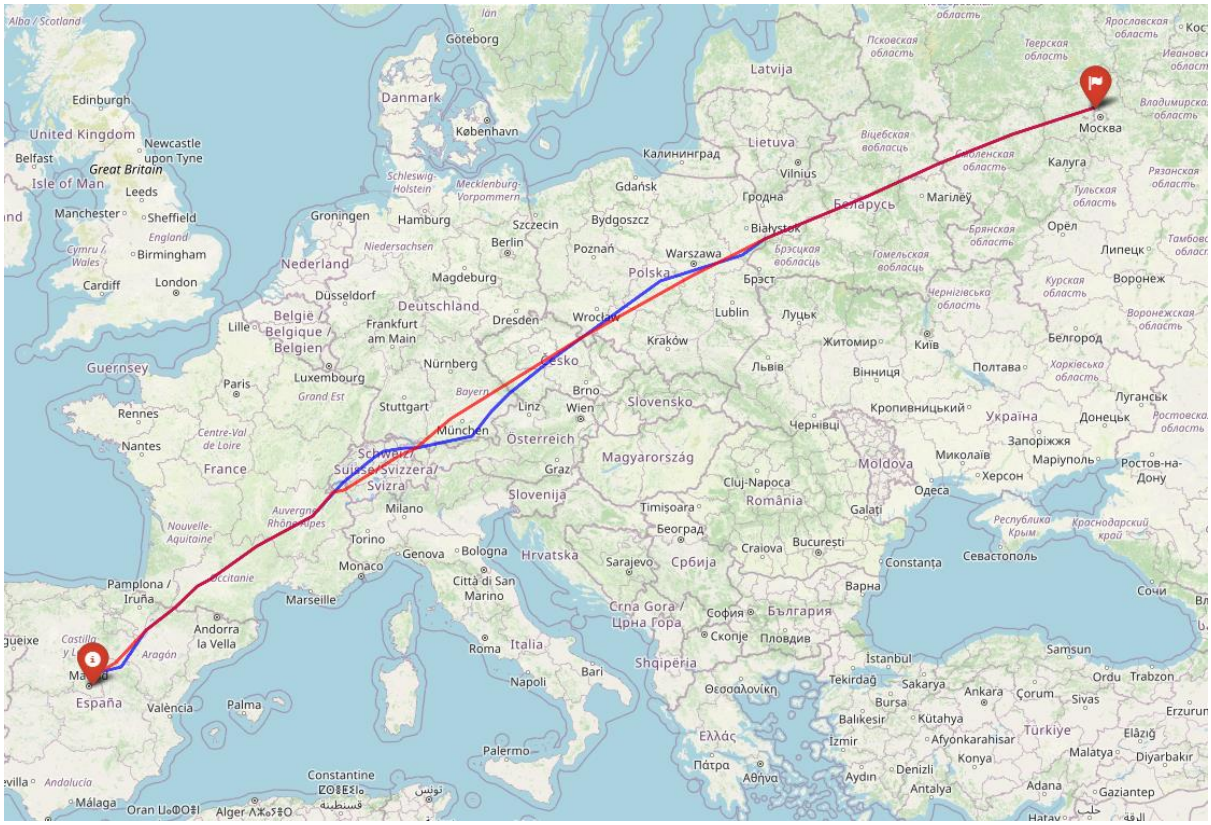

Figura 2.3 Mapa de ejemplo de ruta planificada y ruta real

Es importante destacar que en ningún momento se menciona que este conjunto de datos contenga únicamente vuelos comerciales. El carácter de los vuelos no es el objetivo del proyecto puesto que el análisis se realiza sobre todos los vuelos que operan en los aeropuertos españoles.

2.2 Depuración de la Base de Datos

La depuración de los datos incluye dos partes esenciales para obtener una base de datos limpia: el filtrado de los vuelos que interesan para este trabajo y el filtrado y transformación de los datos disponibles de esos vuelos que vamos a utilizar posteriormente.

2.2.1 Filtrado de los vuelos

En primer lugar, una vista preliminar del contenido de los ficheros de vuelos pone de manifiesto numerosas inconsistencias.

Después de analizar el conjunto de datos original, observamos algunas características atípicas, como, por ejemplo, rutas anormalmente cortas o de muy escasa duración. Debido a esto, se tuvo que revisar varias veces el conjunto de datos en detalle, en un proceso muy repetitivo y laborioso que solo era automatizable para subconjuntos de vuelos. Se analizaron esas anomalías o inconsistencias identificando esos vuelos anómalos y planteando debates sobre qué hacer con ellos.

Estas inconsistencias incluyen vuelos duplicados, datos anómalos o que no nos van a servir para hacer el análisis final o de los que no podemos obtener las características necesarias.

Para realizar el correcto filtrado de esas inconsistencias comparamos, para cada vuelo, los datos que tenemos tanto en el archivo que contiene las rutas planificadas como en el archivo que contiene las rutas reales en un proceso paralelo. A continuación, se describen las distintas inconsistencias que hubo que resolver.

En primer lugar, analizando los datos con los que contábamos a priori vemos que existen vuelos cuyo aeropuerto de salida y aeropuerto de llegada era el mismo. Estos vuelos, los cuales denominamos vuelos circulares, fueron eliminados puesto que la información que nos iban a aportar esas rutas carecía de valor para el análisis en cuestión. En total eliminamos 4.219 vuelos circulares.

En segundo lugar, identificamos que algunos aeropuertos tenían, en el valor que corresponde al aeropuerto de salida o al aeropuerto de llegada, el valor *'ZZZZ'*. Investigando, se descubrió que *'ZZZZ'* es un código ICAO especial utilizado para indicar que un aeropuerto no tiene un código ICAO designado. Este código se utiliza con frecuencia en los planes de vuelo cuando el aeropuerto de salida o destino no es un aeropuerto oficialmente designado con un identificador ICAO. Esto puede ser el caso de helipuertos, pequeños campos de aviación u otros lugares que no están formalmente listados [11]. Estos vuelos también fueron eliminados. En total encontramos 202 vuelos de este tipo.

En tercer lugar, se identificaron algunos vuelos cuya trayectoria real tenía menos de 3 puntos. Estos vuelos, considerados como vuelos cancelados no aportaban ninguna información interesante acerca de un vuelo real del cual no se tienen datos fiables. Únicamente había 25 vuelos con menos de 3 puntos.

En cuarto lugar, el último campo de la trayectoria real, el cual contenía el valor del aeropuerto de llegada, debía coincidir con el aeropuerto de salida. Sin embargo, existen vuelos en los que el aeropuerto de este último punto no coincide con el código ICAO del aeropuerto de llegada. Estos vuelos tenían en este último campo el valor del código ICAO de otro aeropuerto, por lo que fueron considerados como vuelos desviados y también fueron eliminados del conjunto de datos. Encontramos 1.518 vuelos desviados que fueron eliminados.

En quinto lugar, existían vuelos que consideramos raros o sin sentido, considerando que la información que aportaban podía alterar los resultados. Estos vuelos eran los que empezaban o llegaban antes de lo previsto con una diferencia de tiempo notable. Los vuelos eliminados, en este caso, son los vuelos que cumplían 2 condiciones: salían al menos 20 minutos antes de lo planeado y llegaban al menos 1 hora antes de lo que indicaba la ruta planificada. Estos vuelos adelantados fueron un total de 250.

En sexto lugar, analizando los puntos de las trayectorias de una forma más detenida, identificamos que hay algunos puntos en los que se duplicaba el *Datetime* (fecha-hora) de la coordenada y que, además, al menos un valor de la latitud, longitud o altitud no coincidía con los valores del otro punto con el mismo *Datetime*. Estos vuelos fueron eliminados puesto que indican que, en el mismo instante de tiempo, la aeronave estuvo en 2 puntos espaciales distintos. Finalmente, encontramos 202 con *Datetimes* duplicados.

En séptimo lugar, recordemos que los vuelos que están recogidos en esta base de datos incluyen datos desde el 25/03/2018 hasta el 28/10/2018. Existen vuelos que estaban planeados para llegar el 25/03/2018 a una hora temprana pero que al analizar la ruta real se comprueba que llegaron a finales del día anterior. Estos vuelos se han eliminado al ser considerados fuera del rango de la base de datos. De la misma forma, existen vuelos que estaban planeados para llegar el 28/10/2018 pero que finalmente llegaron el día siguiente. Al igual que los anteriores, estos vuelos se eliminaron por estar fuera del rango que indicamos en la base de datos. Se encontraron 193 vuelos fuera de este rango.

Por último, existen vuelos duplicados en la base de datos original. Esto sucede en el caso de que un vuelo salga un día, pero llegue al día siguiente ya sea en la ruta real o en la ruta planificada. Al estar la base de datos organizada por días, ese vuelo está repetido en ambos días. En estos casos las rutas planificadas del vuelo coincidían al igual que el código del vuelo y los aeropuertos de salida y de llegada. Por ello se plantea el debate de qué hacer con estos vuelos. Finalmente, la decisión fue quedarnos con los datos del vuelo cuya ruta real, que era la que marcaba la diferencia, tuviera más puntos o coordenadas en su trayectoria. Por lo tanto, eliminamos un total de 29.869 vuelos duplicados.

Tras filtrar todos estos vuelos pasamos de tener 1.134.813 vuelos en la base de datos original, a 1.098.335 vuelos, por lo que se han eliminado un total de 36.478 vuelos. Dentro de estos vuelos, y sabiendo que se ha seguido el orden de filtrado que se ha descrito en el proceso anterior, podemos saber la distribución, según el tipo de inconsistencia que presentaba, de los vuelos eliminados:

1. Vuelos circulares, es decir, vuelos cuyo aeropuerto de salida y de llegada es el mismo: 4.219 vuelos eliminados.
2. Vuelos con *'ZZZZ'* como valor del código del aeropuerto de salida o del aeropuerto de llegada: 202 vuelos eliminados.
3. Vuelos cancelados, es decir, vuelos cuya trayectoria real tiene menos de 3 puntos o coordenadas: 25 vuelos eliminados.
4. Vuelos desviados, es decir, vuelos cuyo último punto de la ruta real no contiene el código ICAO del aeropuerto de llegada: 1.518 vuelos eliminados.
5. Vuelos que salían y llegaban con bastante adelanto: 250 vuelos eliminados.
6. Vuelos con valores de *Datetimes* duplicados, pero con distintos valores de latitud, longitud o altitud en las coordenadas de la trayectoria real: 202 vuelos eliminados.
7. Vuelos cuya fecha de llegada en la ruta real no corresponda a las fechas que se delimitan en la base de datos, es decir que llegaron el día anterior a la primera fecha (25/03/2018) o el día posterior a la última fecha (28/10/2018): 193 vuelos eliminados
8. Vuelos duplicados porque comenzaron en una fecha, pero llegaron al día siguiente y se han agrupado en ambas fechas: 29.869 vuelos eliminados.

2.2.2 Filtrado y transformación de los datos

Además de los vuelos que hemos filtrado para limpiar el conjunto de datos original, necesitamos decidir cuáles son los datos que vamos a necesitar para la realización del proyecto, así como las unidades en las que queremos que estén medidos.

Basado en la estructura original de los datos la intención es obtener un *Dataframe* final que tenga las siguientes columnas (véase la Figura 2.4):

- Código del vuelo.
- Fecha o día en el que se realizó el vuelo.
- Código ICAO del aeropuerto de salida.
- Código ICAO del aeropuerto de llegada.
- Trayectoria planificada
- Trayectoria real.

flight_id	date	dep_airport	arr_airport	plan_route	real_route
AFL2529AA76304073	25/03/2018	LEMG	None	(4D points)	(4D points)

Figura 2.4 Ejemplo de vuelo Madrid-Moscú del Dataframe después de limpiar los datos

Para ello se crea una estructura de diccionario en la que cada elemento representa un único vuelo con sus datos ya formateados. En la estructura del diccionario (clave, valor) la clave para cada vuelo debía ser única. La primera idea era utilizar el código de vuelo. Sin embargo, este código de vuelo no es único por lo que no servía como clave. Por ello, se utiliza como clave una cadena de texto que contiene el código de vuelo unido a un hash de la ruta planificada de ese vuelo. El valor del diccionario contenía para cada columna indicada anteriormente su valor correspondiente.

Las columnas del código de vuelo, el código ICAO del aeropuerto de salida y el código ICAO del aeropuerto de llegada se recogen directamente ya que esos valores no necesitan ningún formateo anterior. Sin embargo, el alcance de este proyecto son exclusivamente los aeropuertos de España, por lo que en caso de no ser un aeropuerto de España el valor de esa columna del código del aeropuerto de salida o de llegada es *None*.

Para identificar si el aeropuerto pertenecía a España se obtuvo una lista con los aeropuertos españoles, con sus nombres y sus códigos ICAO que se extrajo haciendo *Web Scraping* de una página web [12].

La fecha se indicó a partir del *Datetime* del último punto de la ruta real. Es decir, que la fecha a la que se asigna ese vuelo es la fecha en la que aterrizó dicho vuelo. Era importante definir una fecha de esta forma puesto que había vuelos que estaban planificados para llegar un día y llegaron el día anterior o posterior.

Las trayectorias, tanto real como planificada, tuvieron que ser formateadas ya que el valor que se importaba por defecto de los datos originales era completamente ilegible. Por ello, se identifica cada valor separado por el carácter ':' y tal y como se indica en la Figura 2.5 se separan en cuatro componentes, las cuatro piezas de información importantes:

- El primer valor es un *string* que se corresponde con el *Datetime* de la coordenada.
- El cuarto valor separado por ':' se corresponde con la altitud medida en pies dividido por 100. Por ello se realiza la conversión necesaria para pasarlo a metros.
- El último valor contiene tanto la latitud como la longitud en formato DMS (*Degrees*, *Minutes*, *Seconds*). Por ello se realiza la conversión de estas 2 coordenadas espaciales a grados en formato decimal.

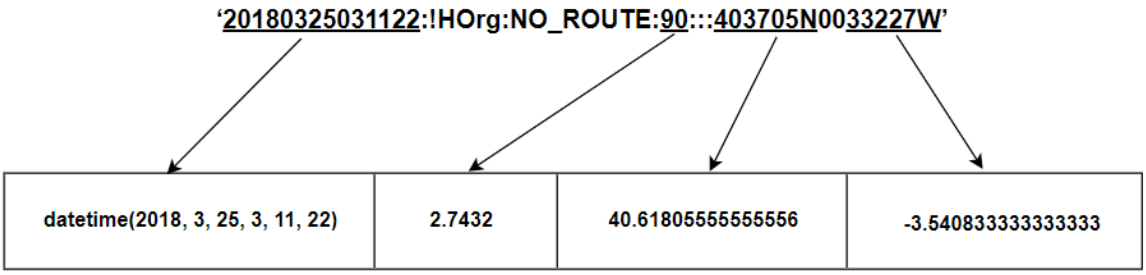

Figura 2.5 Formateo de una coordenada a Datetime, Altitud, Latitud y Longitud

Cabe recordar que, como se mencionó anteriormente, en el primer y en el último punto de cada ruta se encuentra el código ICAO del aeropuerto de salida o de llegada respectivamente, aunque este no sea un valor que guardemos en las trayectorias finalmente.

Por lo tanto, cada trayectoria real y planificada constará de una lista de puntos. Cada punto será una lista de 4 valores, que son (*Datetime*, Altitud, Latitud, Longitud).

Una vez tenemos el diccionario completamente limpio y con los valores en el formato deseado, convertimos este diccionario a una estructura de *Dataframe* con las columnas mencionadas anteriormente.

Este *Dataframe* es el resultado de la limpieza de todos los datos originales. Contiene los datos de 1.098.335 vuelos de 46 aeropuertos españoles distintos entre el 25/03/2018 y el 28/10/2018.

2.3 Cálculo y extracción de las características a evaluar

Antes de poder aplicar el algoritmo Mapper es necesario calcular las características que vamos a utilizar como datos de entrada en este algoritmo. Estas características

son las desviaciones entre la trayectoria real y planificada y el retraso o llegada antes de tiempo del vuelo.

2.3.1 Cálculo del retraso

El retraso del vuelo se refiere a la diferencia temporal entre la hora a la que el vuelo llega en la realidad y la hora estipulada en la que el vuelo debería llegar. Por lo tanto, para calcularlo para cada vuelo que está representado en una fila del *Dataframe* generado anteriormente, necesitamos tomar los últimos puntos de la ruta real y de la ruta planificada. El retraso será la diferencia entre el *Datetime* de este último punto de la ruta real menos el *Datetime* del último punto de la ruta planificada.

Por lo tanto, el retraso podrá tener valores tanto positivos, indicando que el vuelo ha llegado más tarde de lo que estaba planeado; como negativos, indicando que el vuelo se ha adelantado a la hora de llegada planeada.

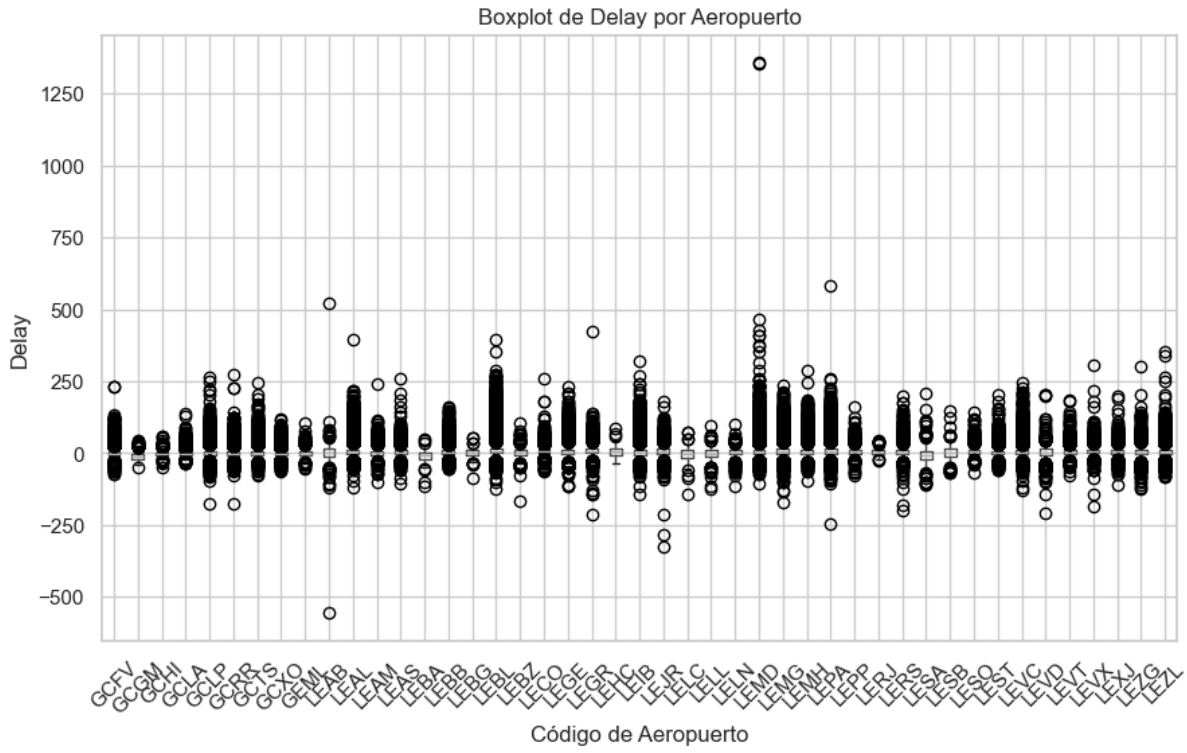

Figura 2.6 Boxplot de los retrasos de los vuelos agrupados por aeropuerto

Los valores obtenidos de los retrasos de los vuelos están representados en la Figura 2.6 en la que podemos ver cómo se obtienen valores tanto positivos, que indican el retraso de los vuelos; como negativos, que indican el adelanto de algunos vuelos. La caja de los *boxplots* paralelos representados en la Figura 2.6 es casi inapreciable por la cantidad de datos atípicos presentes. Hay que destacar que esta es la distribución

de los retrasos por aeropuerto durante el periodo considerado de 217 días, de ahí la gran variabilidad y datos extremos observados.

2.3.2 Cálculo de la desviación entre rutas

La desviación de la ruta es la diferencia en una medida de distancia de lo que se aleja la trayectoria real de la trayectoria planificada. Así pues, en la Figura 2.3 se puede apreciar cómo ambas trayectorias no siguen el mismo camino, aunque tengan el mismo origen y destino. Por tanto, la desviación mide cuánto se diferencian ambas rutas.

A diferencia del cálculo del retraso, el proceso que engloba el cálculo de la desviación entre rutas es bastante más complejo y presenta nuevos retos que afrontar. Como parte de estos retos veremos que es necesario sincronizar las rutas en tiempo, interpolar para poder calcular la distancia entre rutas en los mismos puntos y, finalmente, calcular efectivamente estas distancias.

2.3.2.1 Sincronización de las rutas por normalización del progreso espacial

Para calcular la distancia entre las rutas queremos saber cuánto difieren ambas en longitud si las superponemos. El primer problema que surge es que una ruta comienza antes que otra y, por lo tanto, las distancias se ven afectadas por el tiempo.

Imaginemos que queremos calcular la distancia de un punto de la ruta de un vuelo de Madrid a Moscú en un instante de tiempo concreto. Puesto que ambas rutas no han empezado a la vez y no llegan a la vez puede que en ese instante de tiempo el avión de una ruta se encuentre en un punto cercano a Múnich y en la otra, debido al retraso, se encuentre en un punto cercano a Barcelona.

En este caso podemos ver que la distancia que vamos a obtener en este punto va a ser muy grande y no está reflejando correctamente cuánto se ha desviado la ruta debido a que el retraso está influyendo enormemente en la posición del avión en ese instante y, por tanto, en esa distancia.

Imaginemos otro caso en el que el mismo vuelo sale con una hora de retraso. Eso significa que la distancia que se toma con los puntos que suceden en la primera hora de la ruta planificada van a estar comparándose siempre con el punto del aeropuerto de Madrid puesto que el avión no ha salido todavía en la ruta real.

En estos casos se demuestra que el retraso es determinante y que es directamente proporcional a la distancia si se calculara sin tenerlo en cuenta. Sin embargo, para calcular las distancias entre las rutas queremos calcularla usando su posición relativa independientemente de su duración real y del retraso, positivo o negativo, que tengan.

Por tanto, si se calculan las distancias entre la trayectoria real y la planificada con los datos originales, se corre el riesgo de sobreestimarlas. Consideremos, por ejemplo, la trayectoria planificada (en azul) y real (en naranja) de la Figura 2.7, representando

la altitud del avión. Las trayectorias son muy similares y están desfasadas. Si no tenemos en cuenta el desfase, la distancia entre ellas va a ser mayor de lo que se correspondería con la realidad ya que en los primeros instantes de tiempo el avión planificado empezaría a volar mientras que el real estaría todavía en tierra, inflando artificialmente la distancia entre ellos.

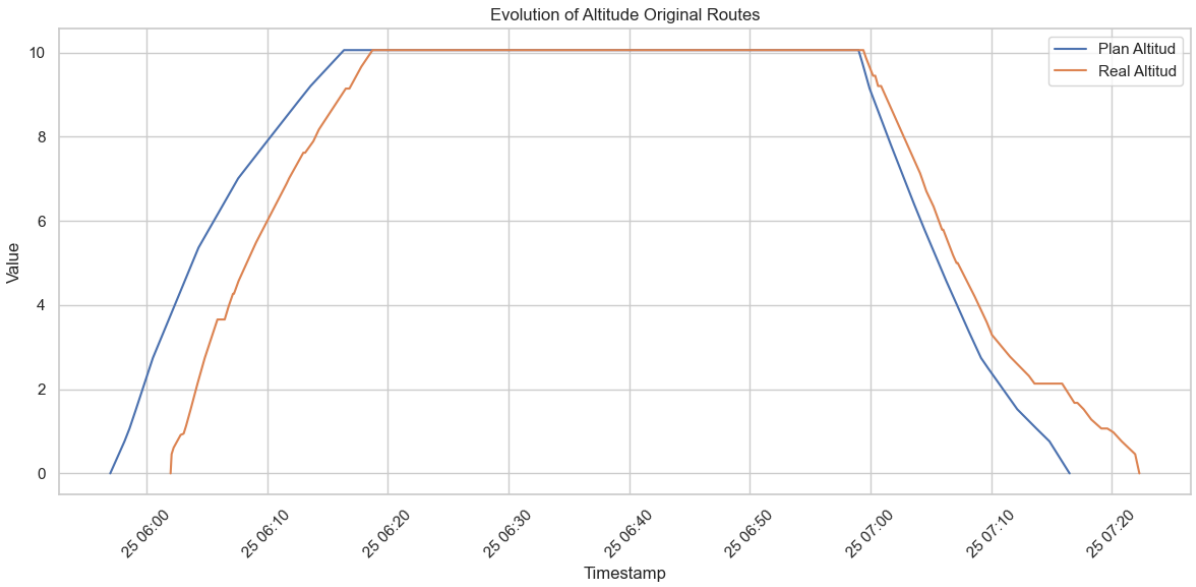

Figura 2.7 Evolución de la altitud con tiempos originales

Este hecho vuelve a ser patente en la Figura 2.8 en la que se representan los primeros puntos de las rutas real y planificada de un vuelo en el plano tiempo-latitud. En ellas podemos ver que, si calculamos la distancia entre las latitudes de las rutas, estas se ven enormemente influenciadas por el tiempo. Sin embargo, ambas gráficas adoptan formas y valores muy similares, aunque en distintos instantes de tiempo. De la misma forma, podemos ver qué pasa exactamente lo mismo para la gráfica de la longitud en la Figura 2.9.

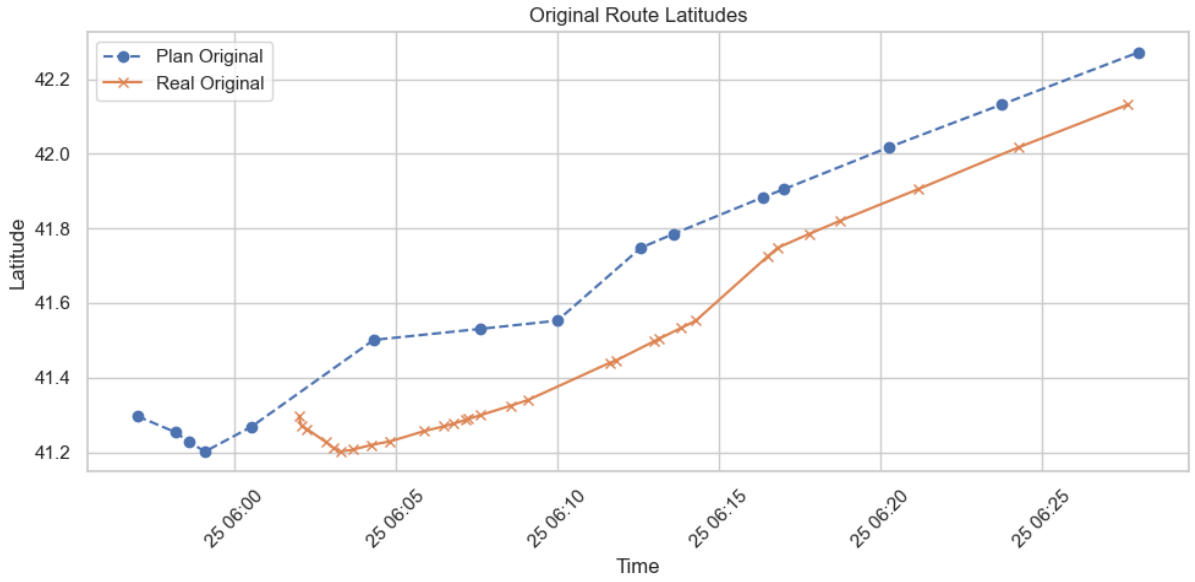

Figura 2.8 Evolución de la latitud con tiempos originales

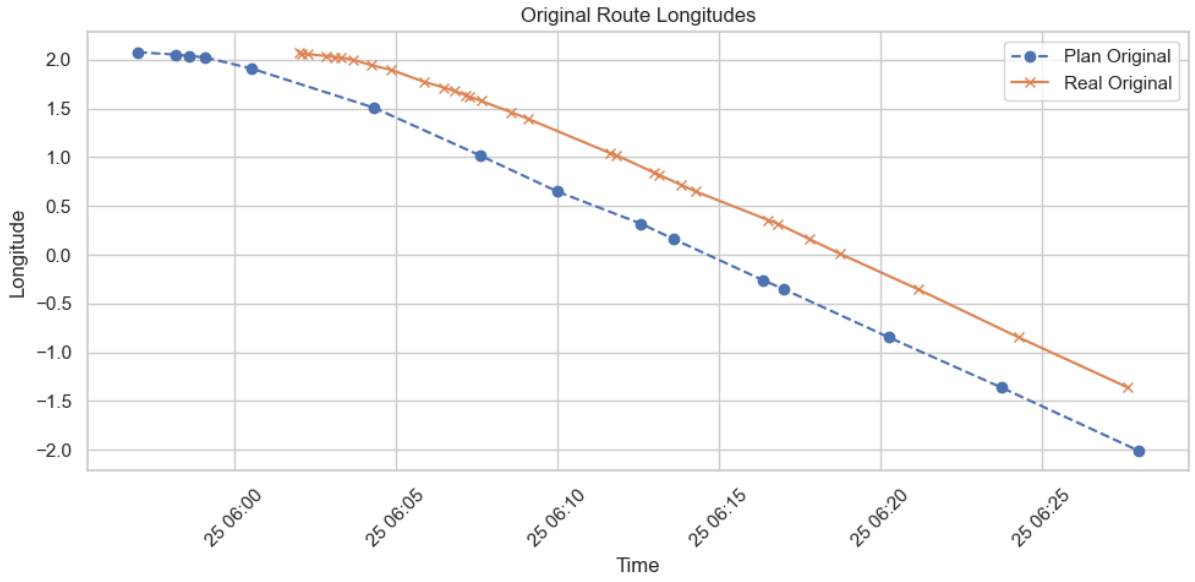

Figura 2.9 Evolución de la longitud con tiempos originales

Por ello, es necesario sincronizar los *Datetimes* de forma que, a la hora de calcular las distancias, estas no se vean afectadas por el retraso. Para solucionar este problema se propuso una primera idea de buscar una sincronización lineal de la ruta y posteriormente otra sincronización que dependiera del progreso espacial de la ruta.

2.3.2.1.1 Sincronización lineal de las rutas

La idea de la sincronización lineal consiste en deformar la escala temporal de la ruta real para que empiece y acabe exactamente igual que la planificada. Para ello se realiza una interpolación lineal de los tiempos de la ruta real basados en el primer y el último instante de tiempo de ambas rutas.

Supongamos que los valores de tiempo son t para la trayectoria planificada y s para la trayectoria real. Se trata de trasladar los puntos s, con $s_1 \leq s \leq s_2$, al intervalo (t_1, t_2) en los que se mueven los puntos t de la trayectoria planificada. La interpolación lineal trasladaría cada valor de s a un valor de t dado por:

$$t = t_1 + \frac{(t_2 - t_1)}{(s_2 - s_1)} (s - s_1) \quad con \quad t_1 < t < t_2 \quad y \quad s_1 < s < s_2$$

Esta sincronización mantiene las proporciones temporales originales entre sus puntos. Sin embargo, al hacer esta sincronización se está asumiendo que el avión recorre su ruta a velocidad constante en el tiempo.

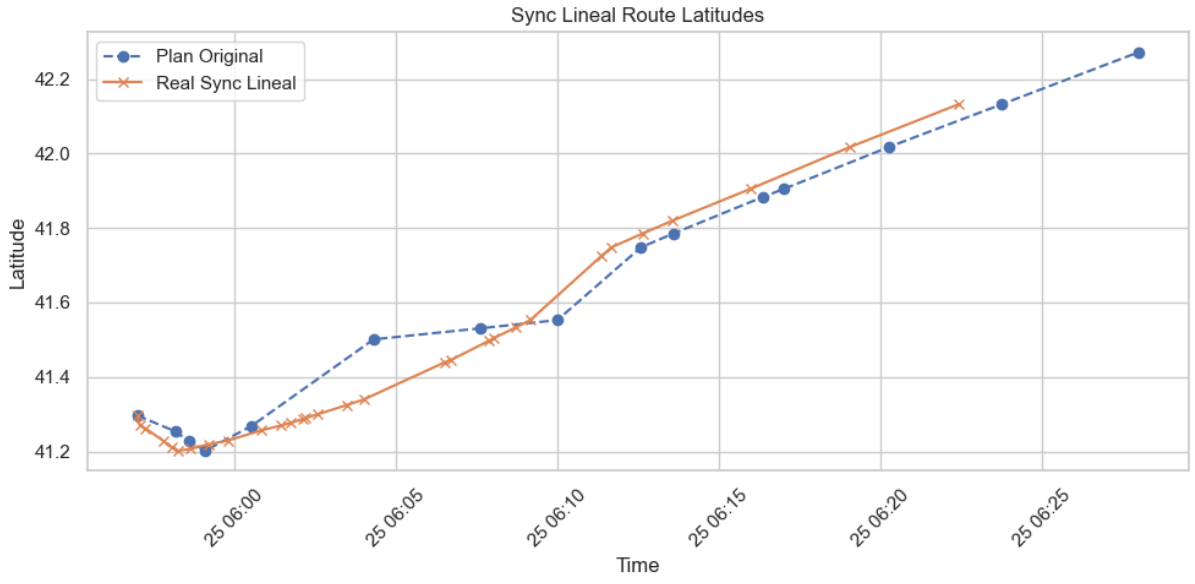

Figura 2.10 Evolución de la latitud con sincronización lineal de los tiempos

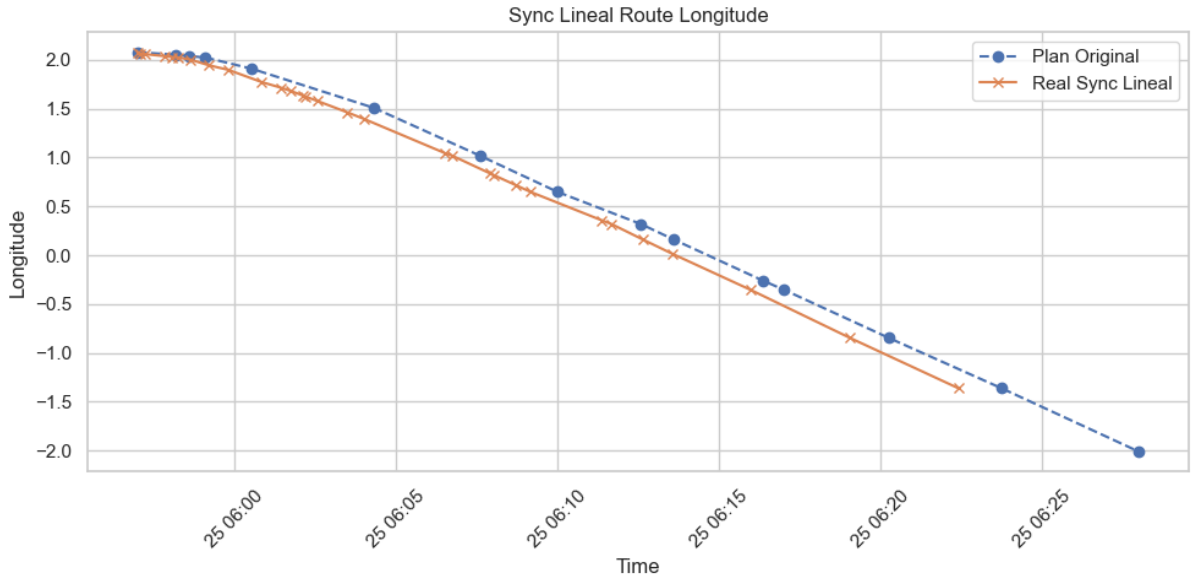

Figura 2.11 Evolución de la longitud con sincronización lineal de los tiempos

Al realizar esta sincronización se puede ver en la Figura 2.10 y Figura 2.11 que las distancias se acortan notablemente debido a la sincronización de los tiempos. Además, se puede verificar que ambas comienzan en el mismo instante de tiempo.

Es importante destacar que solo se están representando los primeros puntos de las rutas para una mejor visualización de esta diferencia, pero si mostráramos las rutas completas también se cumpliría que terminarían en el mismo instante de tiempo.

De esta forma podemos verificar que sincronizar los instantes de tiempos de las rutas es muy útil para obtener distancias más reales y que reflejan mejor la desviación entre las rutas.

2.3.2.1.2 Sincronización por progreso espacial de las rutas

La idea de sincronizar los tiempos de las rutas en base al progreso, parte de la idea de que los vuelos no son homogéneos y que los puntos que tenemos de las rutas no están igualmente espaciados. Dependiendo del instante en el que se encuentre el vuelo, estos puntos tienen una mayor o menor densidad. Por lo tanto, puede que haya tramos del vuelo en los que se muestren muchos más puntos que en otros.

Por ello surge la idea de realizar esta sincronización dependiendo del progreso espacial de las rutas. En este caso sincronizamos las rutas por posición a lo largo del trayecto independientemente de la velocidad del vuelo o de los tiempos reales.

La forma de realizar esta sincronización es en primer lugar calcular el progreso a lo largo del trayecto de forma que dependiendo de sus coordenadas 3D se calcula el progreso del vuelo en ese instante. Este progreso viene representado por un valor entre 0 y 1 que indica el tanto por uno de la ruta que se ha realizado en ese punto.

En segundo lugar, tomando cada punto de la ruta real y su progreso se le asigna a esta ruta real un nuevo *Timestamp* (valor de tiempo) interpolado desde la ruta planificada. De esta forma ambas rutas comparten el mismo marco temporal.

En este caso partimos de una ruta planificada y otra ruta real. Para cada punto de cada ruta se calculará la distancia recorrida en 3 dimensiones entre un punto y el anterior punto de la ruta. Esta distancia se suma a lo que ha recorrido en distancia hasta el punto anterior. El cálculo explícito de esta distancia en 3 dimensiones se explica en el apartado 2.3.2.3.

Para cada punto tendremos un valor de distancia acumulada en 3D. Para calcular la progresión de cada punto se divide la distancia acumulada entre la distancia total. Siendo S_i la progresión en cada punto, esta se calcula como:

$$S_i = \frac{dist.acumulada\ hasta\ el\ punto\ i}{dist.\ Total}$$

Esta progresión para cada punto es un valor entre 0 y 1 se calcula tanto para la ruta planificada como para la ruta real.

Posteriormente, calcularemos el nuevo valor del tiempo realizando una interpolación para obtener un nuevo *Timestamp* en la ruta real a partir de la progresión en cada punto de la ruta real (s) y de la progresión de cada punto en la ruta planificada con n puntos $(s_i, i = 0, \cdots, n)$ y los tiempos de la ruta planificada con n puntos $(\tau_i, i = 1, \cdots, n)$.

$$\tau(s) = \tau_k + \frac{\tau_{k+1} - \tau_k}{s_{k+1} - s_k}(s - s_k)$$

donde k es el índice tal que

$$s_k \leq s \leq s_{k+1}$$

Para ilustrarlo con un ejemplo, supongamos que, calculamos para una ruta planificada de 3 puntos la progresión de cada punto. Esta se calcula como la distancia que lleva recorrida el vuelo en ese punto entre la distancia total del vuelo y obtenemos los valores de progresión que representan el porcentaje de distancia recorrida en ese punto:

$$s_{plan} = [0.0, 0.5, 1.0]$$

Además, cada punto de la ruta planificada tiene asignado un *Timestamp*, que en este caso serían en segundos:

$$\tau_{plan} = [0, 1200, 2400]$$

Posteriormente, y dados los puntos de la ruta real, calcularíamos la progresión de cada punto real de la misma forma. Imaginemos que para una ruta real de 6 puntos hemos obtenido las progresiones:

$$s_{real} = [0.0, 0.25, 0.4, 0.6, 0.75, 1.0]$$

Para obtener el nuevo *Timestamp* que asignaríamos a un punto de la ruta real, calculamos este *Timestamp* de la forma que se hace en el siguiente ejemplo para cada punto.

En este caso, suponemos que cogemos el punto de la ruta real cuya progresión es $s_{real} = 0.25$.

En primer lugar, identificamos entre qué 2 puntos de las progresiones de la ruta planificada se encuentra este punto de la ruta real.

$$s_0 = 0 \leq 0.25 \leq s_i = 0.5 \rightarrow k = 0$$

Y calculamos el tiempo utilizando la fórmula que hemos descrito antes,

$$\tau(0.25) = 0 + \frac{1200 - 0}{0.5 - 0} \times (0.25 - 0) = \frac{1200}{0.5} \times 0.25 = 2400 \times 0.25 = 600\,s$$

Finalmente sumaríamos esos segundos al *Timestamp* original de la ruta real obteniendo el nuevo *Timestamp* sincronizado.

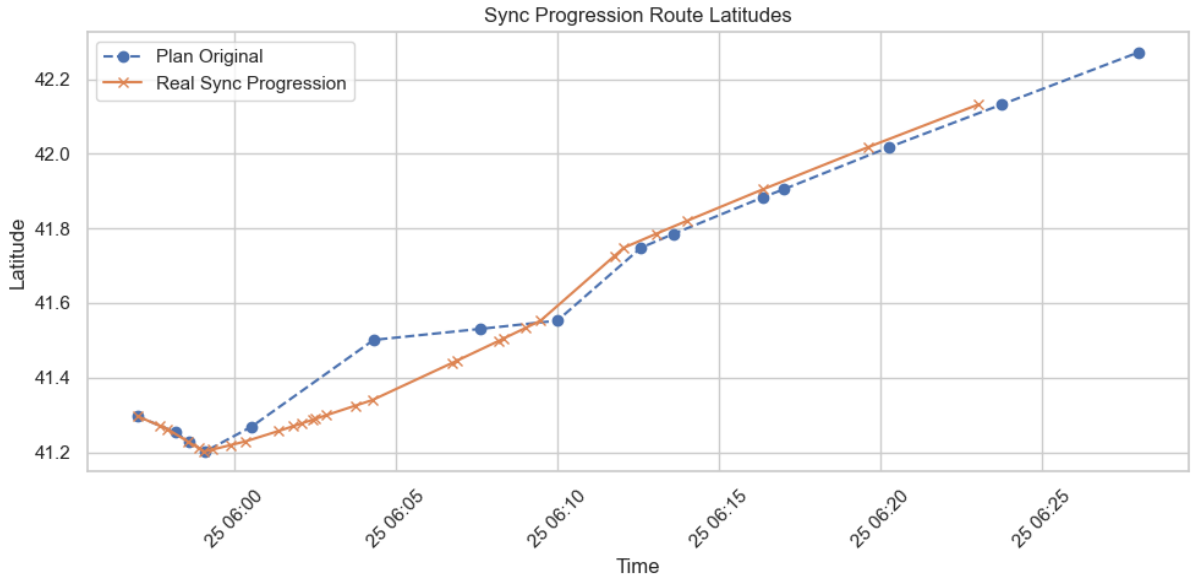

Figura 2.12 Evolución de la latitud con sincronización por progresión

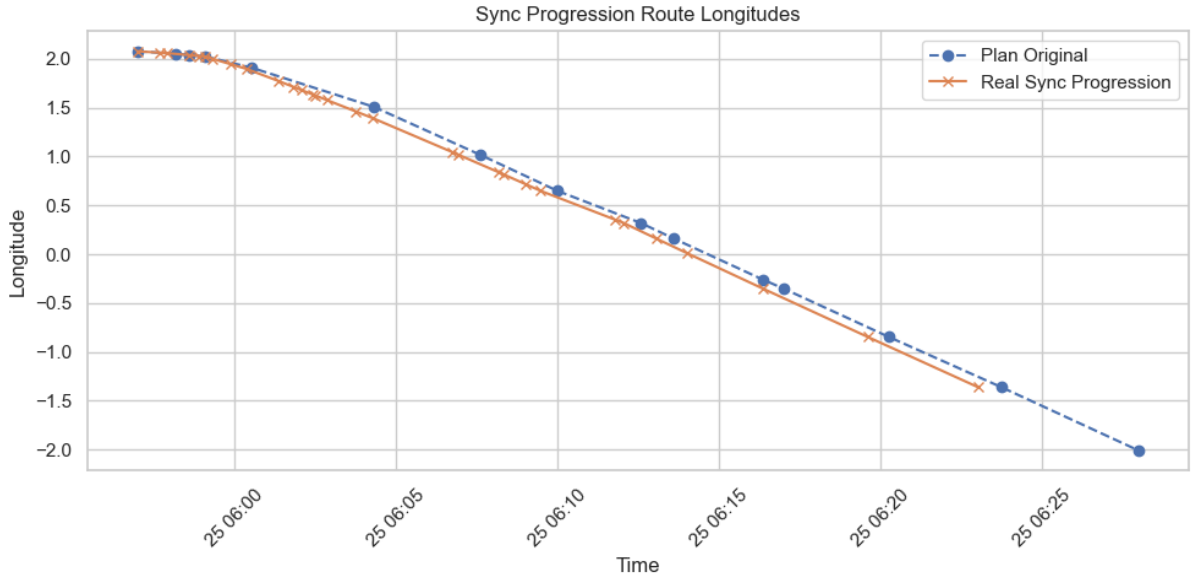

Figura 2.13 Evolución de la longitud con sincronización por progresión

Esta sincronización consigue ser más realista en el caso que queramos comparar el comportamiento físico, ya que queremos calcular únicamente la desviación de las rutas.

Además, se puede comprobar en la Figura 2.12 y Figura 2.13 que la distancia que se refleja para cada característica de la ruta es aún menor que con la sincronización lineal de las rutas propuesta anteriormente.

2.3.2.2 Interpolación de las rutas

Una vez que tenemos los tiempos sincronizados y ambas rutas situadas en el mismo marco temporal, podemos fijarnos en los puntos efectivos que tenemos de cada ruta. De esta forma, tal y como se aprecia en los histogramas de la Figura 2.14 y Figura 2.15 podemos ver que el número de puntos que se tiene en la ruta planificada no es el mismo que en la ruta real.

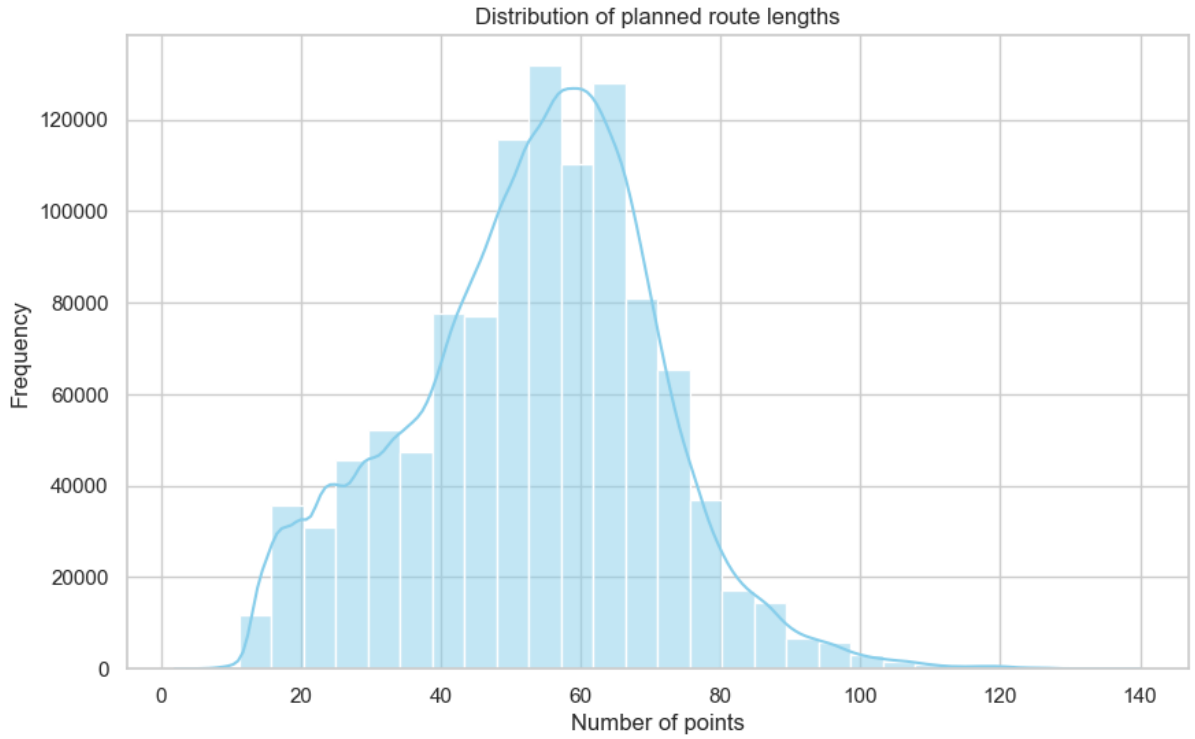

Figura 2.14 Distribución de los puntos de la ruta planificada

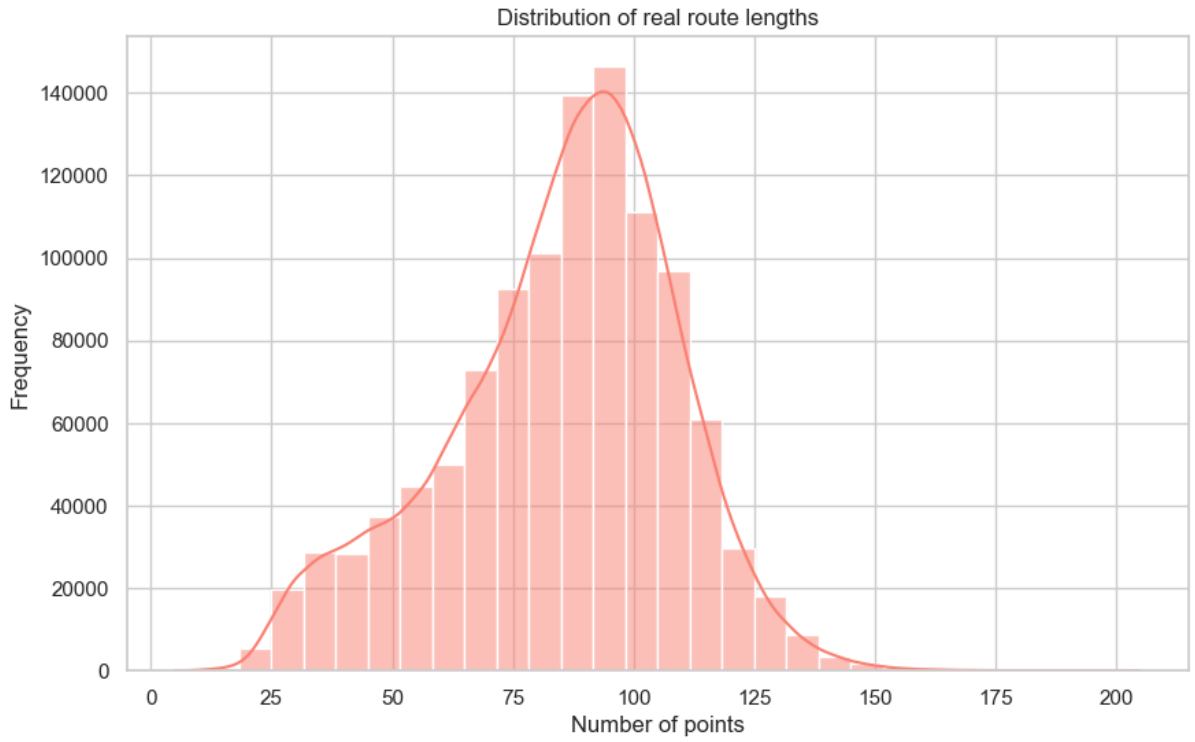

Figura 2.15 Distribución de los puntos de la ruta real

En estos casos las rutas reales tienen más puntos que las rutas planificadas lo que hace que no se pueda calcular las distancias punto a punto y calcular así la desviación sin ninguna transformación de por medio.

Por esta razón surge la idea de realizar una interpolación. La idea consiste en reconstruir 2 rutas nuevas interpolando latitud, longitud y altitud para una serie de tiempos. Esta serie de tiempos engloba la unión de todos los *Timestamps* que están en la ruta real y en la ruta planificada.

Para cada punto de *Timestamp* se asignará un nuevo valor de latitud, longitud y altitud a la ruta. Por lo tanto, como ambas rutas comienzan y terminan en el mismo instante de tiempo, tras la sincronización anterior, ambas tendrán el mismo valor de latitud, longitud y altitud para el comienzo y el final de las rutas.

Sin embargo, para cada punto de *Timestamp* intermedio se realiza una interpolación que calculará el valor de latitud, longitud y altitud dependiendo de los 2 puntos, uno por encima y otro por debajo de ese *Timestamp* en cada ruta.

Figura 2.16 Ejemplo de nuevos puntos de rutas interpoladas

Como podemos ver en la Figura 2.16, los puntos azules y rojos representan los puntos originales de las rutas antes de ser completadas por interpolación, para la ruta planificada y real, respectivamente. En la misma figura, los puntos verdes y amarillos representan los puntos de las rutas interpoladas para la ruta planificada y real, respectivamente.

En este gráfico podemos ver que existe una mayor cantidad de puntos que hacen referencia a las rutas interpoladas (puntos verdes y amarillos) que de puntos que hacen referencias a las rutas originales (puntos azules y rojos).

El resultado de esta interpolación es la obtención de 2 rutas que tienen el mismo número de puntos con los mismos valores de *Timestamp*. Esto permite que el cálculo de las distancias punto a punto se pueda realizar de forma directa.

2.3.2.3 Cálculo de la distancia de las rutas

Finalmente, tenemos una ruta planificada y una ruta real interpoladas en la que los *Timestamps* de ambas se han sincronizado por lo que están en el mismo marco temporal.

Para calcular la desviación entre ambas, primero calculamos la distancia punto a punto entre ambas rutas. Debido a que las distancias son en 3 dimensiones utilizamos primero la distancia de Haversine [13] para calcular la distancia horizontal (latitud/longitud).

La distancia de Haversine se emplea para calcular la ruta más corta sobre la superficie de una esfera, en este caso la utilizamos sobre la superficie de la Tierra. Se calcula a partir de las coordenadas de longitud y latitud de dos puntos y viene medida en metros:

$$d = 2R \sin^{-1}\left(\sqrt{\left(\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1) * \cos(\varphi_2) * \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)\right)}\right)$$

En esta fórmula:

1. d es la distancia entre los dos puntos en línea recta sobre la superficie de la esfera
2. R es el radio de la esfera, en este caso el de la Tierra. R = 6,371 Km.
3. $\varphi_1 \, y \, \varphi_2$ son las latitudes de los dos puntos en radianes. La latitud se mide desde el ecuador (que es 0°), con valores positivos hacia el norte y negativos hacia el sur.
4. $\lambda_1 \, y \, \lambda_2$ son las longitudes de los dos puntos en radianes. La longitud se mide desde el meridiano de Greenwich (0°), con valores positivos hacia el este y negativos hacia el oeste.

Una vez tenemos calculada la distancia de Haversine en ese punto entre ambas rutas, calculamos la diferencia de altitud entre los 2 puntos como la diferencia de la altitud en dicho punto de la ruta planificada menos la altitud en el punto de la ruta real. Esta será la distancia vertical.

La distancia final en ese punto será la raíz cuadrada del sumatorio de ambas distancias, horizontal y vertical, elevadas al cuadrado, o lo que es lo mismo, la aplicación del teorema de Pitágoras.

Finalmente, la desviación global entre rutas la calcularemos como el sumatorio de las distancias en cada punto temporal.

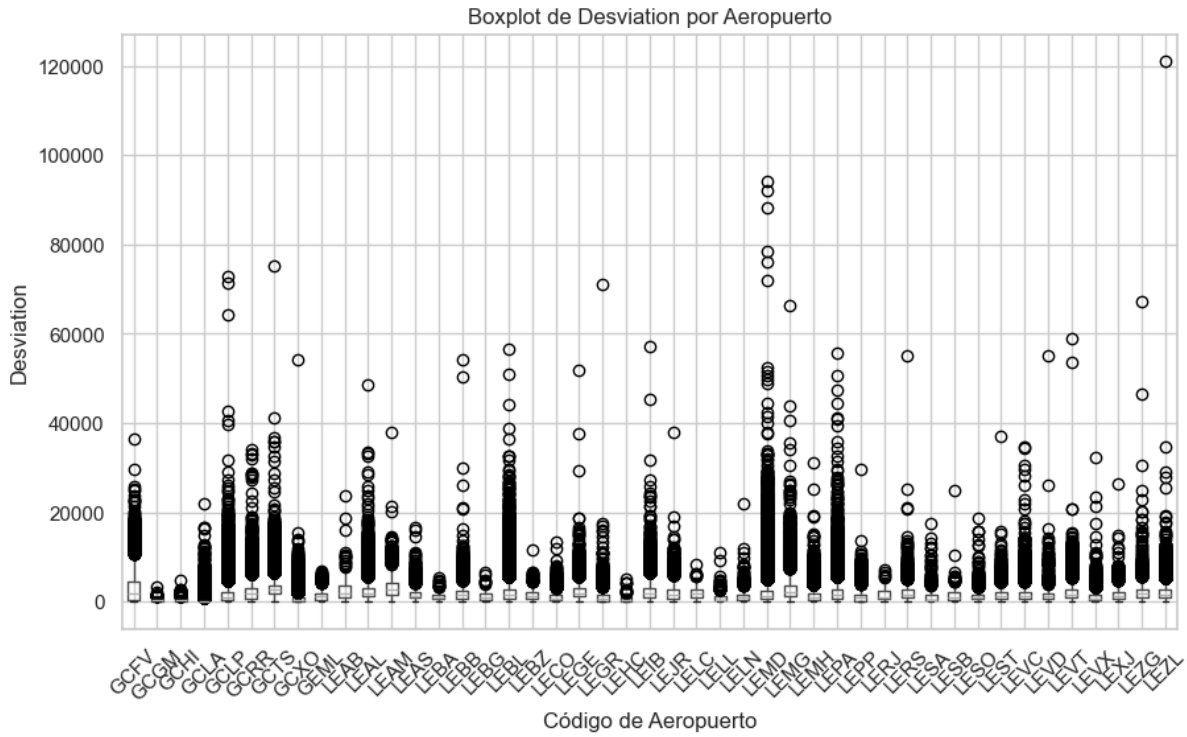

Figura 2.17 Boxplots de la desviación de los vuelos agrupados por aeropuerto

En la Figura 2.17 están representadas las desviaciones entre rutas de los vuelos medidas en metros y agrupadas por aeropuertos.

Se puede apreciar la alta variabilidad de los datos y la abundancia de datos atípicos que hacen que casi no se aprecien las cajas de los diagramas. Hay que tener en cuenta que se están representando desviaciones entre rutas por aeropuerto en un total de 217 días.

Cabe destacar que por cada vuelo se tiene un aeropuerto de salida y un aeropuerto de llegada. Puesto que posteriormente se quieren agrupar los vuelos por aeropuertos se generará una entrada en el fichero de datos para el vuelo para el aeropuerto de salida y otro para el aeropuerto de llegada.

Dado que hemos filtrado y eliminado anteriormente los aeropuertos que no son españoles, no se generará un valor de vuelo para esos aeropuertos.

Estos datos, además, se dividen distinguiendo que el aeropuerto sea de salida o de llegada.

Es decir, en caso de que tengamos un vuelo Madrid – Barcelona, se añadirán las características del vuelo una vez para el aeropuerto de Madrid en los datos de vuelos de salida y otra para el aeropuerto de Barcelona en los datos de vuelos de llegada.

Sin embargo, en caso de que el vuelo sea Madrid – Moscú, las características del vuelo únicamente se añadirán con el código del aeropuerto de Madrid para los datos de los vuelos de salida.

3 Algoritmo Mapper y Resultados

En este capítulo se explica qué es el algoritmo Mapper que surge como fusión del Análisis Topológico de Datos y de la teoría de grafos. Esta es una introducción no rigurosa desde el punto de vista matemático que tiene el objetivo de que sea fácilmente entendible en una primera lectura.

Además, también se incluyen los resultados obtenidos a partir de la aplicación de este algoritmo al conjunto de datos procesado y preparado en el capítulo anterior.

3.1 Análisis Topológico y Algoritmo Mapper

El volumen de datos que se está generando en cualquier ámbito está creciendo y, cada vez, a un ritmo más acelerado. Estos datos son el combustible para muchas tecnologías modernas como pueden ser el procesamiento del lenguaje natural o el reconocimiento de imágenes [14].

Sin embargo, cuantos más datos hay, mayor es la complejidad de analizar esos datos. Ahí es donde el Análisis Topológico de Datos (TDA) resulta útil. En lugar de trabajar directamente con los datos en bruto, el TDA busca extraer la forma subyacente de los datos.

3.1.1 Análisis Topológico

El Análisis Topológico de Datos es una rama emergente y cada vez más influyente de la Ciencia de Datos. Este análisis es reconocido por ser capaz de extraer información relevante en conjuntos de datos de alta dimensión, en donde los métodos tradicionales no son tan efectivos.

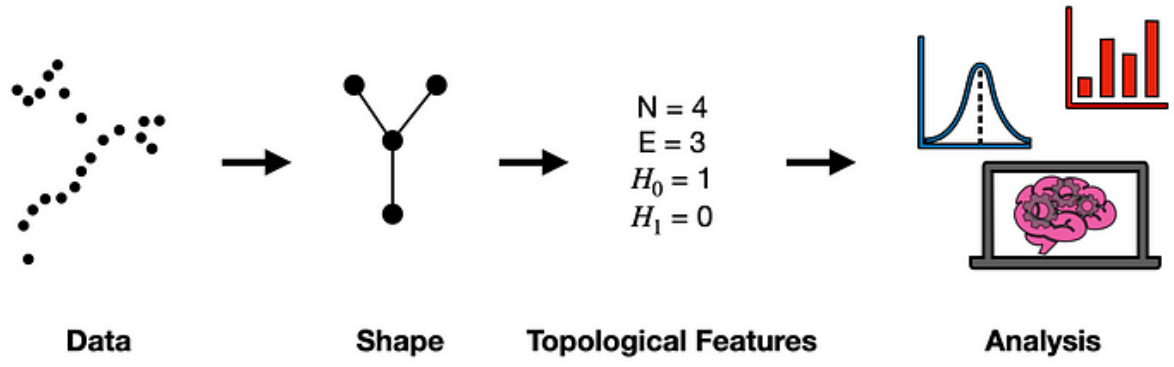

Figura 3.1 Proceso básico de Análisis Topológico de Datos. Fuente: [14]

A diferencia de otros enfoques estadísticos, el Análisis Topológico de Datos se centra en aquellas propiedades del conjunto de datos que permanecen invariantes ante transformaciones continuas. Es por esa razón que este análisis permite revelar la

estructura global y las relaciones internas en conjuntos de datos de alta dimensión ofreciendo una visión más completa de su organización como muestra la Figura 3.1.

Este proceso comienza con un conjunto de datos que podríamos pensar como una nube de puntos en N dimensiones en el que cada dimensión representa una variable medida sobre cada unidad de observación.

A partir de esa nube de puntos necesitaríamos generar formas y de esas formas obtener los rasgos topológicos de ese conjunto de datos. Esos rasgos podrían ser, por ejemplo, contar el número de nodos y aristas de un grafo, partes conexas o contabilizar el número de 'agujeros' en los datos.

Finalmente, esos rasgos topológicos son los que se utilizan para hacer un análisis. Este análisis podría incluir clasificar los datos según las estadísticas de sus rasgos topológicos o emplearlos como variables de entrada para un modelo de aprendizaje automático.

Además, existen 2 enfoques distintos acerca de la manera en la que generar esas formas que permiten obtener los rasgos topológicos. El primero de ellos es la Homología persistente. La homología persistente permite caracterizar los datos a través de los agujeros, un rasgo fundamental de la topología que tiende a ser robusto al ruido. Estos agujeros se detectan en distintas dimensiones a partir de la transformación de los datos en un complejo simplicial, donde los puntos cercanos se conectan a través de bolas [15]. Sin embargo, este enfoque no es en el que nos centramos en este trabajo.

El segundo enfoque es el algoritmo Mapper. Este algoritmo acaba traduciendo el conjunto de datos en un grafo interactivo. Además, es un algoritmo ideal para analizar y visualizar datos de alta dimensión [16].

3.1.2 Algoritmo Mapper

Un grafo es una imagen compuesta por puntos a los cuales denominamos nodos. Estos nodos están conectados a través de otros elementos, los cuales denominamos aristas.

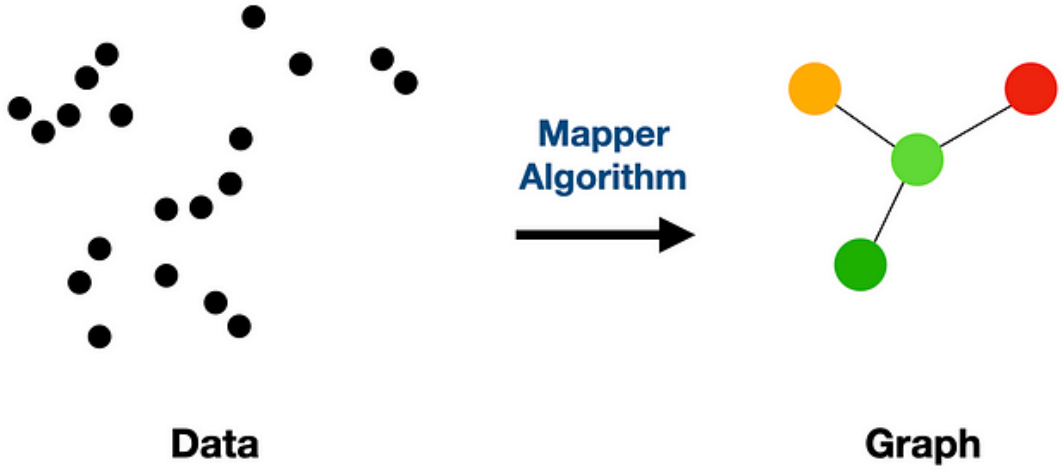

Figura 3.2 Idea general del algoritmo Mapper de generar un grafo a partir del conjunto de datos [17]

En el caso de que tuviéramos un conjunto de datos original en 2 dimensiones generaríamos finalmente un grafo como representa la Figura 3.2. Sin embargo, no hay un límite para la dimensionalidad de los datos de entrada.

Para llegar a obtener el grafo final, el algoritmo contiene una serie de pasos que parten del conjunto de datos original en dimensión *N*. Como hemos mencionado anteriormente podemos entender este conjunto de datos como una nube de puntos en un espacio N-dimensional con tantas dimensiones como variables tienen estas observaciones.

El siguiente paso consiste en proyectar los datos en una dimensión más baja. A esta proyección se le suele denominar función lente o filtro. Esta función mapea los datos de entrada de mayor dimensión a una representación de menor dimensión.

Existen muchas formas de realizar esa reducción de dimensión. Puede ser tan simple como descartar todas las variables de los datos menos una o una estrategia de reducción de dimensión más sofisticada como puede ser *Principal Componente Analisis* (PCA), *t-Distributed Stochastic Neighbor Embedding* (t-SNE) o *Uniform Manifold Approximation and Projection* (UMAP), entre otros. La Figura 3.3 muestra la distribución de la frecuencia de uso de distintos métodos de reducción de la dimensionalidad en trabajos en los que se utiliza el algoritmo Mapper.

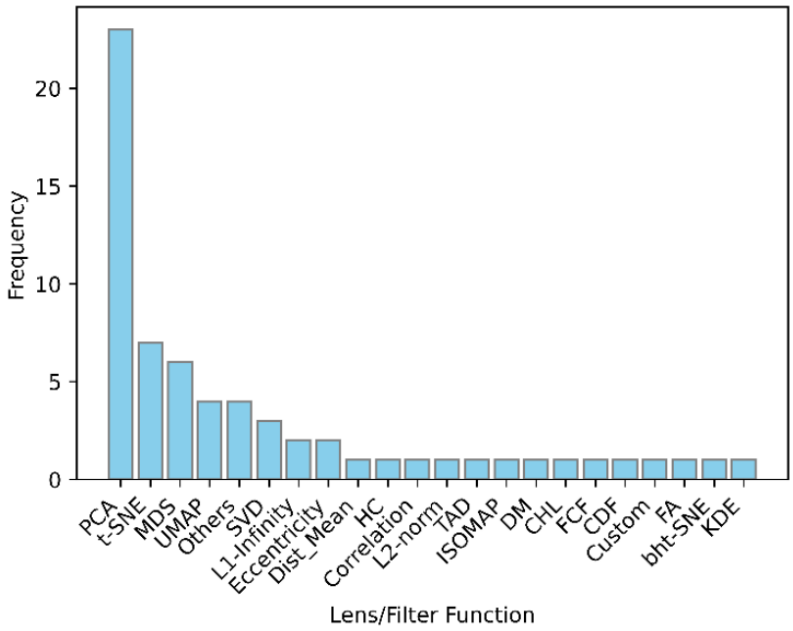

Figura 3.3 Distribución de uso de funciones lente o filtro usadas en análisis Mapper [18]

Para la realización de este proyecto se ha utilizado PCA como función de filtro. PCA es la función lente más utilizada en proyectos de análisis en los que se utiliza el algoritmo Mapper [18].

El Análisis de Componentes Principales (PCA) es una técnica de Ciencia de Datos que sirve para reducir el número de dimensiones de un conjunto de datos tratando de mantener la mayor parte posible de la información contenida en ellos (entendiendo por información la variabilidad de los datos). Las nuevas dimensiones obtenidas por esta técnica no representan las dimensiones originales, sino que son combinaciones lineales de las variables originales que maximizan la variabilidad contenida en las mismas. Estos componentes capturan la mayor cantidad de información posible del conjunto de datos original.

En la Figura 3.4 podemos ver como los datos, que originalmente se encuentran en 2 dimensiones se proyectan en 1 sola dimensión.

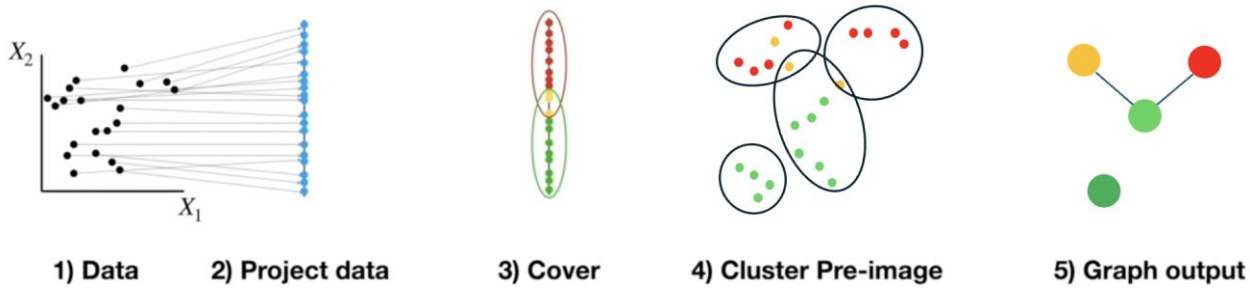

Figura 3.4 Resumen de los pasos del algoritmo Mapper. Elaboración propia inspirada en [17]

Posteriormente, se define una cobertura para los datos proyectados. Esta cobertura se define como la creación de subconjuntos solapados de los puntos de datos proyectados. Por lo tanto, un porcentaje de los datos formarán parte de ambos subconjuntos.

En la Figura 3.4 los datos proyectados en una única dimensión se dividen en 2 subconjuntos indicados por los colores rojo y verde. Además, aquellos datos que pertenecen a ambos subconjuntos, es decir, la intersección de ambos subconjuntos, se indica con el color amarillo.

Una vez tenemos hecha la cobertura podemos ver que tenemos el conjunto de datos dividido en subconjuntos. Sin embargo, cada punto sigue siendo un punto en el espacio de datos original. En el siguiente paso se toma un subconjunto en la cobertura (como puede ser el subconjunto del círculo rojo en la Figura 3.4), se ve a qué puntos corresponden en el conjunto de datos original y se aplica un algoritmo de agrupamiento o *clustering* como puede ser *k*-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) o Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN).

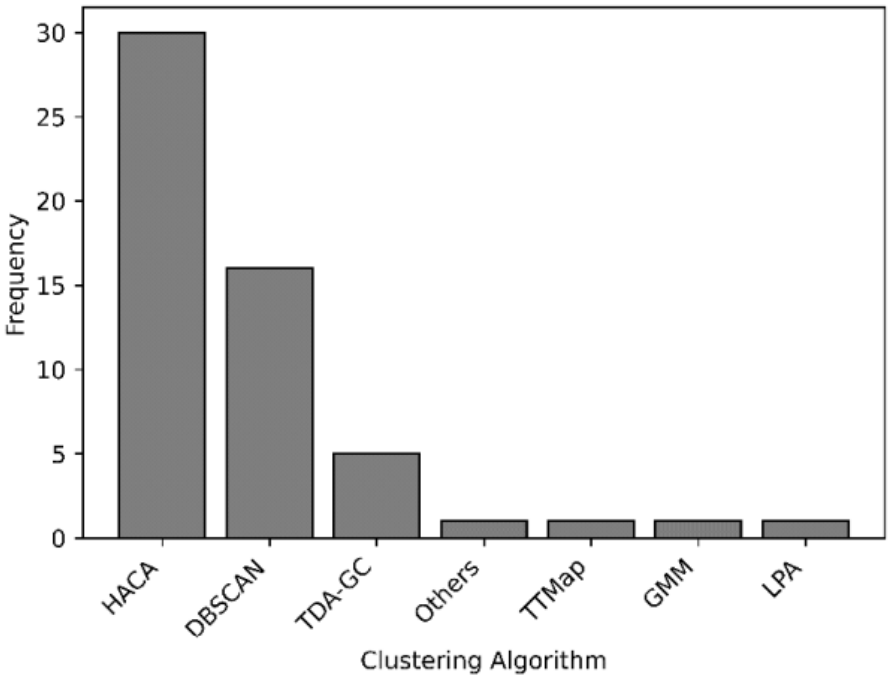

Figura 3.5 Distribución de uso de algoritmos de clustering usados en análisis Mapper [18]

En nuestro proyecto hemos utilizado 2 algoritmos de *clustering* distintos: DBSCAN y HDBSCAN. DBSCAN es un algoritmo con eficiencia probada en la aplicación del algoritmo Mapper [18] y HDBSCAN es una variación del primero. En la Figura 3.5 podemos ver cuáles son los algoritmos de *clustering* más utilizados en los trabajos en los que se utiliza el algoritmo Mapper. Algunos de estos algoritmos de *clustering* son *Hierarchical Agglomerative Cluster Analysis* (HACA), DBSCAN o *Topological Data Analysis - Graph Clustering* (TDA-GC).

DBSCAN es un método de *clustering* adecuado para buscar patrones de agrupación en el espacio físico. Para ello, este algoritmo agrupa los puntos que están más cercanos respecto a alguna métrica, comúnmente se utiliza la distancia euclidiana. Además, para este algoritmo, cada clúster definido contendrá un número mínimo de puntos.

Hierarchical DBSCAN (HDBSCAN) es un método de *clustering* que calcula la jerarquía de todos los clústeres de DBSCAN para un mínimo de puntos. Luego utiliza una estabilidad basada en el método de extracción para encontrar cortes óptimos en la jerarquía, por lo que produce una solución plana [19].

El proceso de *clustering* se repite por cada subconjunto definido después de aplicar la cobertura. Fijándonos en la Figura 3.4 podemos ver que existen grupos de datos que comparten observaciones.

Por último, se construye un grafo basado en los grupos generados en el paso anterior. En ese grafo, cada grupo generado en el paso anterior se representa como un nodo y, en el caso de que dos nodos tengan algún punto en común, se conectan ambos nodos.

3.2 Resultados

Recordemos que la base de datos obtenida finalmente contenía para cada vuelo sus características de retraso y su desviación entre trayectoria real y planificada. Para obtener unos análisis interesantes utilizando el algoritmo Mapper estos vuelos se han agrupado de 2 formas distintas.

En un primer caso, se propone el agrupamiento más general posible. En él se agrupan los vuelos por aeropuerto, considerando todo el intervalo de tiempo de 217 días.

En el segundo caso, también se agrupan los vuelos por aeropuerto, pero esta vez se separan por meses. Por lo que se obtiene un grafo por cada mes, agrupando los aeropuertos.

3.2.1 Preparación del algoritmo Mapper

Tanto si consideramos todos los vuelos a la vez como si consideramos los de cada uno de los meses, agrupamos los vuelos por aeropuertos. Para ello juntamos los vuelos en los que coincida el aeropuerto y sobre estos aeropuertos calculamos las características descriptivas que se utilizarán finalmente como los datos de entrada del algoritmo Mapper.

Estas características son la media, la mediana, la desviación típica y el rango intercuartílico. Se calculan tanto para la desviación entre rutas como para el retraso, así como para los datos de vuelos de salida como para los datos de los vuelos de llegada. Por lo tanto, tenemos un total de 16 características distintas para cada aeropuerto, las cuales podemos ver en la Figura 3.6.

deviation_mean_dep	deviation_median_dep	deviation_std_dep	deviation_iqr_dep	delay_mean_dep	delay_median_dep	delay_std_dep	delay_iqr_dep
2344.814885	1390.821004	2560.690100	2457.702446	-0.917071	-2.966667	14.346604	14.833333
274.946616	224.047083	223.247584	264.088464	-4.829581	-6.000000	9.376121	9.333333
359.374056	322.307876	194.276413	222.184474	-2.742581	-3.691667	8.053626	9.133333
801.299793	289.824771	1422.706129	334.878341	-1.787814	-2.883333	8.525683	8.600000
1572.813124	579.780411	2242.388306	2129.628499	0.384889	-0.983333	12.052679	12.900000
deviation_mean_arr	**deviation_median_arr**	**deviation_std_arr**	**deviation_iqr_arr**	**delay_mean_arr**	**delay_median_arr**	**delay_std_arr**	**delay_iqr_arr**
3079.005991	3783.673574	2607.658528	4983.504387	-0.404947	-1.616667	11.441832	11.400000
679.957038	616.179451	250.805857	203.840244	-20.294278	-21.025000	8.941157	9.987500
568.341909	479.158419	470.263760	639.448996	-0.622293	-1.083333	8.352575	10.358333
448.432746	145.941698	809.213556	271.185742	0.128403	-1.300000	9.094736	9.658333
1582.045439	1102.756425	1577.934583	1493.780925	2.307241	0.716667	14.049667	15.791667

Figura 3.6 Entrada de las 16 características (media, mediana, desviación típica y rango intercuartílico de la desviación de las distancias entre trayectorias y del retraso para los vuelos de salida y de llegada al algoritmo Mapper.

Sin embargo, estas características están en su escala original por lo que a la hora de usar una técnica de *clustering* las distancias de esos puntos se verán afectados por las unidades en las que se encuentran medidas. Por esa razón, transformamos o escalamos las características utilizando un StandardScaler generando una distribución para cada variable con media ≈ 0 y varianza ≈ 1. Para ello, a cada valor se le resta la media y se divide por la desviación típica de la variable.

La matriz original de 16 variables (características) va a sufrir una primera reducción de dimensionalidad utilizando PCA. Puesto que el algoritmo Mapper realiza el *clustering* en el espacio original, es mejor reducir esas 16 variables originales para evitar la maldición de la dimensionalidad (cuanto mayor es la dimensión, mayores son las distancias promedio entre puntos). La elección de 3 componentes principales viene motivada por la variabilidad explicada que se obtiene con este número de componentes, que supera el 80% de la variabilidad total. Una vez que tenemos la matriz de entrada preparada es necesario definir los parámetros y las técnicas para la ejecución del algoritmo Mapper. En primer lugar, definimos la proyección o reducción de dimensionalidad que se va a utilizar (filtro o *lens*, en la nomenclatura Mapper), que será igual a 1. Es decir, con PCA igual a 3 como base para el *clustering* y filtro con PCA igual a 1 se espera obtener una estructura global simplificada y un *clustering* más expresivo y robusto. Estas reducciones se obtuvieron tras numerosas pruebas ensayo-error con diversos valores. La complejidad de los grafos obtenidos con mucha redundancia en la información mostrada que se traduce en muchos nodos conectados con exactamente las mismas observaciones nos iba dando indicaciones de los mejores valores a utilizar en las reducciones aplicadas.

Hay que tener en cuenta que, en la elección del resto de parámetros, no solo influyen las dimensiones elegidas anteriormente, sino también el número de observaciones (aeropuertos) que se van a agrupar, que no es muy grande, solamente 46.

Seguidamente se definen los parámetros de la cobertura. Estos parámetros son:

- N_cubes: Define el número de cubos en cada dimensión del espacio proyectado. Es el número de cubos en cada dimensión para segmentar el espacio.
- Perc_overlap: Define el porcentaje de solapamiento entre los cubos adyacentes en la segmentación del espacio.

Además, también necesitamos incluir los parámetros del método de clustering que utilicemos. En el caso de DBSCAN:

- Eps: Es el radio o la distancia máxima entre 2 puntos para que se consideren vecinos.
- Min_samples: Define el número mínimo de puntos que debe tener un vecindario para que un punto sea considerado núcleo y pueda formar parte de un clúster.

En el caso de HDBSCAN no requiere un valor para el radio ya que se adapta automáticamente a la densidad local de puntos. Los parámetros en este caso son:

- Min_cluster_size: Indica el número mínimo de puntos que debe tener un grupo para ser considerado como clúster.
- Min_samples: Igual que DBSCAN. Es el número mínimo de puntos que debe haber en el entorno de un punto para ser considerado un núcleo.

Para la ejecución del algoritmo Mapper se ha utilizado la biblioteca KeplerMapper [20] que permite generar un grafo dinámico en formato HTML. A partir de este HTML se puede visualizar las diferencias de los nodos y las características más relevantes del grafo.

3.2.2 Resultados para vuelos agrupados por aeropuerto

Para el primer ejemplo en el que se agrupan los vuelos únicamente por aeropuerto, los valores que se han utilizado en el primer caso en el que se usa DBSCAN son:

- Cover:
  - N_cubes = 12
  - Prec_overlap = 0,25
- DBSCAN:
  - Eps = 2,5
  - Min_smaples = 1

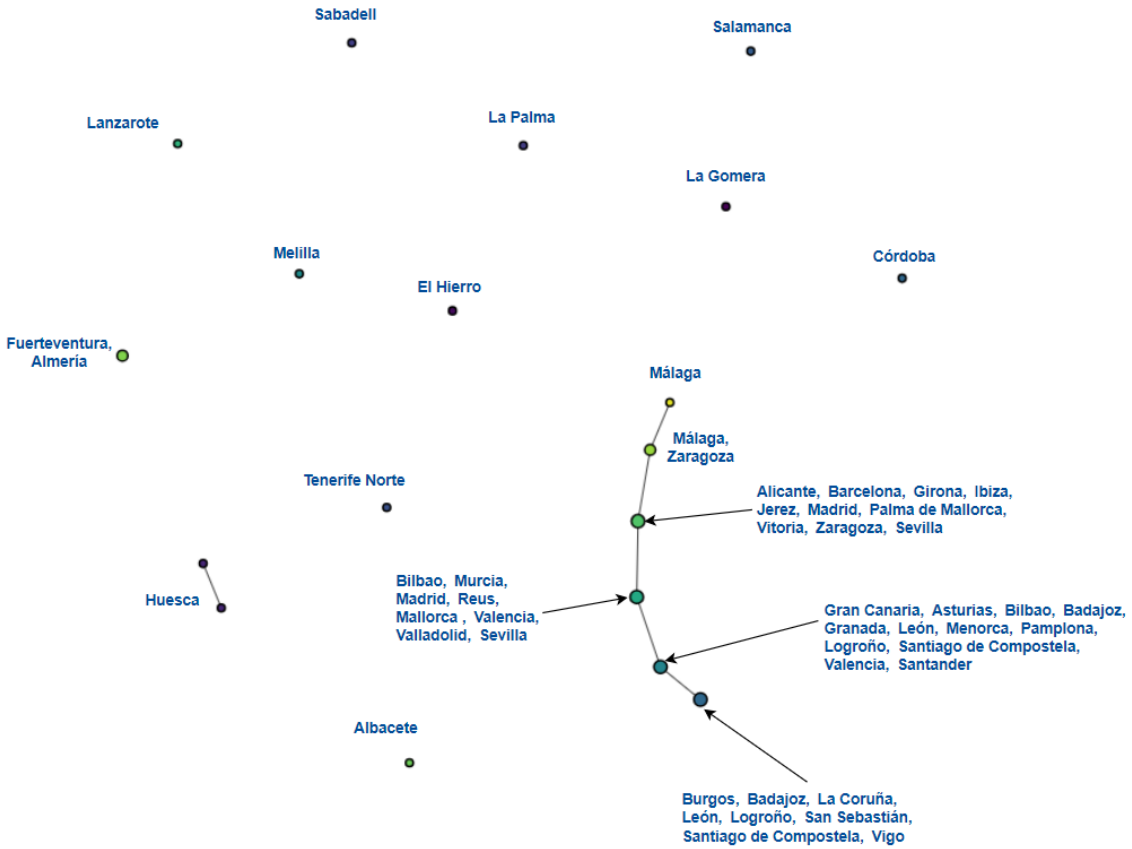

Figura 3.7 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos y utilizando DBSCAN

En la Figura 3.7 podemos ver el grafo resultante de la ejecución del algoritmo Mapper con los parámetros descritos. El grafo tiene 20 nodos y 6 enlaces. Podemos visualizar que existe un tronco principal en el que se encuentran la gran mayoría de los aeropuertos mientras que el resto se encuentran en nodos separados.

Podemos identificar que en el tronco de la rama principal se encuentran la mayoría de los aeropuertos que tienen un perfil similar o incluso con mayor fluctuación aérea. Sin embargo, los aeropuertos pequeños que no están conectados, como El Hierro, La Gomera, Huesca, Melilla o Sabadell están separados del núcleo por lo que pueden ser considerados como *outliers* funcionales.

Si nos centramos un poco más en los aeropuertos de la rama principal, podemos ver como los aeropuertos que más tráfico aéreo se espera que tengan, principalmente en las fechas que engloba el conjunto de datos, se encuentran en el mismo clúster. Este es el caso de, por ejemplo, Madrid, Barcelona o Palma de Mallorca.

Si analizamos un poco más los aeropuertos que no están conectados con la rama principal, vemos que se trata de aeropuertos con menos tamaño y menos tráfico aéreo. Prácticamente todos estos aeropuertos están aislados sin generar más grupos entre ellos y la mayoría son aeropuertos que no tienen mucho tráfico aéreo o en los que sus rutas, dentro del territorio nacional, son más cortas.

Este es el caso de muchos aeropuertos de las Islas Canarias. Los aeropuertos más pequeños de estas islas podemos ver que se encuentran en estos nodos separados. En ellos podemos identificar, debido a los colores en el grafo interactivo, que sus retrasos y desviaciones no son extremadamente grandes. Esto tiene sentido puesto que estos aeropuertos tienen un mayor tráfico entre islas que a destinos más alejados. Al final, en vuelos de distancias cortas es menos probable tener valores de desviación altos.

Sin embargo, los aeropuertos de las islas que más tráfico tienen durante esta época del año sufren mayores valores de retraso, agrupándose con los aeropuertos con más vuelos de la península. También tiene sentido puesto que cuanta mayor sea la distancia recorrida de un vuelo mayor puede ser la desviación en la ruta real con respecto a la planificada.

Al analizar un poco las métricas del grafo que nos proporciona el algoritmo Mapper podemos darnos cuenta de que el grafo está muy poco conectado con una densidad del grafo sumamente baja (0.0316). Además, tenemos hasta 10 nodos con grado 0 que no están conectados y el grado mayor que tenemos en el grafo es de 2 por lo que las máximas conexiones de un nodo son 2.

Por otra parte, podemos ver que algunos aeropuertos forman parte de varios nodos en el grafo de la Figura 3.7. Los dos primeros nodos de la rama principal, empezando por la parte inferior del grafo, tienen en común los aeropuertos de Badajoz, Santiago, León y Logroño. Estos aeropuertos tienen características comunes en cuanto a las variables analizadas se refiere con los aeropuertos no comunes en estos dos primeros nodos.

Utilizando DBSCAN podemos ver que en el grafo se representan todos los aeropuertos en clústeres, independientemente del número de aeropuertos que haya en cada clúster, puesto que así está indicado en el valor min_samples de este algoritmo. Esto nos sirve para identificar una estructura general de los aeropuertos y cómo se agrupan, así como cuáles son los aeropuertos que más se distancian de la mayoría del grupo.

En el segundo ejemplo se siguen utilizando los vuelos agregados por aeropuerto, pero esta vez utilizando HDBSCAN como método de *clustering*. Los parámetros que introducimos en el algoritmo son:

- Cover:
  - N_cubes = 7
  - Prec_overlap = 0,25
- HDBSCAN:
  - Min_cluster_size = 2
  - Min_smaples = 1

El valor del parámetro min_cluster_size de HDBSCAN es el mínimo que permite este algoritmo. El grafo resultante del algoritmo Mapper es el de la Figura 3.8.

Figura 3.8 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos y utilizando HDBSCAN

De este gráfico podemos ver en una primera inspección que el número de nodos que encontramos se reduce a prácticamente a la mitad, pasando de 20 nodos en el primer grafo a 11 en este. Otra característica importante es el número de aeropuertos distintos que tenemos en este grafo en donde vemos una reducción de 45 aeropuertos distintos que teníamos en el grafo con DBSCAN a 23 aeropuertos. Esto se debe también a que este algoritmo no muestra nodos con un solo elemento.

Cabe destacar que estos 23 aeropuertos están englobados todos en los 3 nodos con más aeropuertos del anterior grafo. Por lo tanto, todos estos aeropuertos se encuentran en los 3 nodos del tronco principal del grafo obtenido al ejecutar el algoritmo con DBSCAN. Parece que este grafo con HDBSCAN amplía la imagen de la nube de puntos concentrándose en las zonas en las que hay una mayor densidad de aeropuertos.

Analizando este grafo podemos ver que no hay ninguna rama principal, si no una separación muy clara entre distintos aeropuertos. Podemos destacar que el algoritmo ha juntado los aeropuertos que tienen mayor tránsito según la clasificación realizada por AENA en 2018 [21] que son el de Madrid, Barcelona y Palma de Mallorca. Estos tres aeropuertos son calificados como grupo especial por AENA. Además, los une junto a otros como Jerez y Girona, que pertenecen a distintos grupos en la

calificación hecha por AENA. Jerez es calificado por AENA en el Grupo 2 (aeropuertos con número de pasajeros anuales entre medio millón y dos millones) y Girona en el grupo 3 (aeropuertos con número de pasajeros anuales menor de medio millón). Hay que recordar en este punto que la clasificación hecha por AENA solamente tiene en cuenta el número de pasajeros por año de los aeropuertos, sitúa en un grupo especial a los aeropuertos canarios y crea un grupo especial con los aeropuertos más transitados. Las agrupaciones generadas en este trabajo no tienen en cuenta directamente el tránsito de pasajeros sino las distancias entre las rutas reales y planificadas y los retrasos de los vuelos. Y aunque los retrasos sí pueden estar relacionados con la cantidad de vuelos que operan en el aeropuerto, las diferencias entre trayectorias involucran más factores relacionados con la división del espacio aéreo y su regulación. Por ello, no es de extrañar que estos dos tipos de agrupaciones difieran. Curiosamente, los aeropuertos de Jerez y Girona pertenecen ambos al grupo 2 si nos remitimos al número de pasajeros durante los meses de abril a septiembre de 2022 [21]. No hemos tenido acceso a estos datos concretos de número de pasajeros para 2018.

Por otro lado, también se observa que este algoritmo une aeropuertos geográficamente próximos como pueden ser el de Vigo y La Coruña o el de Bilbao y Santander.

Si analizamos las estadísticas de este grafo podemos ver que sigue siendo un grafo poco conectado, aunque con algo más de densidad (0.727). Sin embargo, el diámetro del grafo o la ruta más larga entre nodos es mucho más corta puesto que no hay una rama principal como existía en el grafo realizado con DBSCAN.

3.2.3 Resultados para vuelos agrupados por aeropuertos y divididos por meses

En este caso, primero dividimos los vuelos por meses. Ya que los vuelos que tenemos en la base de datos van desde el 25/03/2018 hasta el 28/10/2018 hemos agregado únicamente los vuelos con datos de meses completos, es decir, los vuelos de los meses de abril, mayo, junio, julio, agosto y septiembre. Ya que de cada vuelo tenemos una fecha asignada, utilizamos esta variable para filtrar los vuelos en cada caso.

Una vez tenemos los vuelos filtrados por meses, realizaremos el mismo proceso para cada grupo de vuelos. En primer lugar, agrupamos los vuelos por aeropuerto de la forma que lo hicimos en el caso anterior. Posteriormente ejecutaremos el algoritmo Mapper utilizando los mismos parámetros que utilizamos en el caso anterior:

Para el caso en el que usamos DBSCAN como clustering:

- Cover:
  - N_cubes = 12
  - Prec_overlap = 0,25
- DBSCAN:
  - Eps = 2,5
  - Min_samples = 1

En el caso en el que usamos HDBSCAN como clustering:

- Cover
  - N_cubes = 7
  - Prec_overlap = 0,25
- HDBSCAN:
  - Min_cluster_size = 2
  - Min_samples = 1

Figura 3.9 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando DBSCAN, de los vuelos de agosto

Figura 3.10 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando HDBSCAN, de los vuelos de agosto

La Figura 3.9 y la Figura 3.10 podemos ver los grafos que obtenemos después de ejecutar el algoritmo Mapper para el mes de agosto, utilizando como método de *clustering* tanto DBSCAN como HDBSCAN. Además, los distintos grafos del resto de meses se encuentran en el capítulo de Anexos. Las diferencias que hay entre los distintos grafos indican que existen diferencias en las características del tráfico aéreo en los distintos meses que afectan a las agrupaciones hechas de los aeropuertos.

En primer lugar, nos centramos en una visión más global de todas las características de los vuelos, que es la que tenemos en los grafos obtenidos de la ejecución del algoritmo Mapper con DBSCAN como método de *clustering*. En estos grafos podemos ver todos los aeropuertos del conjunto de datos. Los distintos grafos tienen entre 17 y 21 nodos distintos y entre 6 y 10 enlaces entre dichos nodos.

En ellos podemos ver que la forma de los grafos varía según el mes. En los meses de junio, julio, agosto y septiembre podemos ver que los grafos tienen una rama o tronco principal que agrupa aeropuertos caracterizados por tener más tránsito aéreo. Fuera de esta rema principal, se encuentran aeropuertos en nodos individuales aislados o nodos con muy pocos aeropuertos.

Sin embargo, en los meses de abril y mayo podemos ver que hay 2 troncos principales. Uno de ellos engloba los aeropuertos con más tráfico aéreo en estos meses y en él se encuentran los aeropuertos con más tráfico de toda España. En el otro tronco se encuentran algunos aeropuertos isleños (en abril, de los dos archipiélagos españoles y en mayo solamente de las Islas Canarias) junto con otros aeropuertos medianos. En este segundo tronco es destacable la cantidad de aeropuertos de la zona del

noroeste de la península Ibérica que hay: La Coruña, Vigo, León, Burgos, Asturias, San Sebastián.

Cabe destacar que los aeropuertos del grupo especial en la clasificación hecha por AENA se encuentran siempre en las ramas principales y, además, tienden a situarse en el mismo grupo o en grupos enlazados, indicando su parecido en los retrasos y desviaciones debido a su gran afluencia de tráfico aéreo.

Si nos fijamos en los nodos más aislados de estos grafos podemos ver que la mayoría son nodos individuales. En estos nodos se encuentran la mayoría de los aeropuertos pertenecientes a las islas Canarias: El Hierro, La Palma, Fuerteventura, La Gomera, Tenerife Norte. Además, estos aeropuertos se unen en pequeñas ramas en algunos meses como es en el caso de junio, julio o septiembre.

Sin embargo, estos no son los únicos nodos aislados que aparecen. También aparecen otros nodos aislados con aeropuertos considerados de bajo tráfico como son Huesca, Almería, Albacete, Salamanca o Córdoba.

Si nos fijamos en las métricas que obtenemos de estos grafos, mostradas en la Tabla 1, podemos ver que abril y mayo presentan diámetros pequeños (3) con esas 2 ramas principales que aparecen. Además, ambos meses tienen relativamente pocos componentes (10-13), indicando que sus grupos están ligeramente cohesionados, teniendo en cuenta que el número de aeropuertos a agrupar no es muy grande (46).

Los grafos de todos los meses presentan densidades muy bajas, es decir, muestran muy pocos enlaces entre nodos, comparándolos con el máximo de enlaces posible. Julio representa, en este caso, el grafo más disperso, ya que tiene la densidad más baja de todos los grafos (0,0351) y también el grado medio más bajo (0,6316). Además, tiene el máximo número de componentes conectados junto a mayo con 13, lo que indica que tiene muchos nodos aislados o en pares.

En el grafo que representa los vuelos de agosto, hay menos nodos que en el resto de los grafos (17). Además, la densidad en este grafo aumenta (0,0588) y se reduce el número de componentes conectados, alcanzando el mínimo para todos los grafos analizados (9) sugiriendo que el grafo tiene un mayor agrupamiento. También alcanza el máximo diámetro (6) corroborando su mayor compacidad comparado con el resto de los meses.

Finalmente, el grafo que representa los vuelos de septiembre tiene un gran parecido a la tendencia que tiene el mes de junio en las estadísticas con el mismo valor de densidad (0,0476), de grado medio (0,9524) y de número de componentes conectados (11).

Tabla 1 Métricas de los grafos obtenidos con DBSCAN para los vuelos de los meses desde abril hasta septiembre

Mes	**Nodos**	**Enlaces**	**Densidad**	**Grado medio**	**Componentes**	**Tamaño medio de nodo**	**Diámetro**
Abril	19	9	0,0526	0,9474	10	3,105	3
Mayo	21	8	0,0381	0,7619	13	2,714	3
Junio	21	10	0,0476	0,9524	11	3,095	5
Julio	19	6	0,0351	0,6316	13	3,000	4
Agosto	17	8	0,0588	0,9412	9	3,588	6
Septiembre	21	10	0,0476	0,9524	11	3,048	6

A modo de resumen, agrupamos los meses que comparten similitudes clave en los agrupamientos y destacamos las principales características del clúster en la Tabla 2:

Tabla 2 Agrupación de clústeres de los meses por sus estadísticas para los grafos obtenidos con DBSCAN

Clúster	**Meses**	**Características comunes**
C1	Abril, Junio, Septiembre	Alta conectividad, grado medio alto, estructura relativamente estable
C2	Mayo, Julio	Menor conectividad, más fragmentación (más componentes), bajo grado medio
C3	Agosto	Densidad máxima, tamaño de nodo más alto, menos fragmentación, estructura más centralizada

Estos clústeres coinciden con los que se obtienen si aplicamos algoritmos de *clustering*, tales como, el *k*-means o el dendograma, lo que refuerza la robustez de los grupos.

En segundo lugar, nos fijamos en una visión más detallada representada por los grafos que se obtuvieron utilizando HDBSCAN como el algoritmo de *clustering* del Mapper. Al igual que anteriormente, se reduce considerablemente el número de nodos que hay en estos grafos. En estos grafos podemos ver diferencias en el número de nodos con valores entre 10 y 16 nodos y con un número de enlaces entre 1 y 6, por lo que hay una gran diferencia entre estos grafos dependiendo del mes.

Si nos fijamos en la forma de estos grafos podemos ver claramente que tienen varios nodos con pocos aeropuertos y dispersos, sin formar ramas o troncos principales con muchos aeropuertos. Sin embargo, esta regla no se cumple en los meses de julio y agosto en la que los grafos concentran muchos aeropuertos en pocos nodos, aunque mantienen pocos enlaces. Las diferencias de estos dos meses son muy notables.

Podemos ver como la tendencia de juntar aeropuertos que están en el grupo especial de AENA con los 3 aeropuertos con más tráfico (Madrid, Barcelona y Palma de Mallorca) únicamente se mantiene en los meses especiales de julio y agosto. En el resto, el aeropuerto de Madrid y el de Palma de Mallorca no siguen una tendencia clara, aunque Madrid se agrupa con Sevilla en los meses de mayo y abril. Sin embargo, el aeropuerto de Barcelona tiende a unirse en el mismo nodo que Girona en la mayoría de los casos.

Basados las estadísticas que obtenemos de los grafos generados con el algoritmo Mapper utilizando HDBSCAN podemos generar un resumen en la Tabla 3. En él podemos ver que el grafo que representa el mes de abril, utilizando HDBSCAN, presenta varios grupos pequeños en los que se pueden formar ramas de hasta 3 componentes siendo el máximo diámetro de este grafo de 2.

Los grafos de mayo y junio representan redes muy distendidas con muy pocos enlaces (3 y 1 respectivamente) y con alto número de componentes conectados, siendo el máximo en mayo (11).

El grafo que representa los vuelos de julio sigue una estructura bastante similar a los de junio, pero con un enlace más. Una gran diferencia con junio es que el número de aeropuertos que hay por nodo en este grafo es muy superior llegando al máximo de estos grafos (3,900).

El grafo de agosto es el más cohesionado de estos grafos, puesto que es el grafo con mayor densidad (0,0606) y con un alto grado medio (0,6667). Además, aunque el diámetro de este grafo sea bajo (2), es el más alto de estos grafos.

Finalmente, el grafo de septiembre vuelve a degradarse con una densidad notablemente menor (0,0455) y con un valor inferior de diámetro (1) lo que indica que no hay caminos largos.

Tabla 3 Métricas de los grafos obtenidos con HDBSCAN para los vuelos de los meses desde abril hasta septiembre

Mes	**Nodos**	**Enlaces**	**Densidad**	**Grado medio**	**Componentes**	**Tamaño medio de nodo**	**Diámetro**
Abril	16	6	0,0500	0,7500	10	2,750	2
Mayo	14	3	0,0330	0,4286	11	2,429	1
Junio	10	1	0,0222	0,2000	9	2,500	1
Julio	10	2	0,0444	0,4000	8	3,900	1
Agosto	12	4	0,0606	0,6667	8	3,666	2
Septiembre	12	3	0,0455	0,5000	9	2,500	1

Si aplicamos algoritmos de *clustering*, tales como, el *k*-means o el dendograma, obtenemos los clústeres que se detallan en la Tabla 4:

Tabla 4 Agrupación de clústeres de los meses por sus estadísticas para los grafos obtenidos con HDBSCAN

Clúster	**Meses**	**Características comunes**
C1	Mayo, Junio, Septiembre	Muy baja conectividad, grado muy bajo
C2	Julio, Agosto	Tamaño medio alto, menor número de componentes
C3	Abril	Intermedio en todo: densidad, grado, componentes

Si comparamos las diferencias entre el análisis hecho con el algoritmo de clustering DBSCAN y el realizado con HDBSCAN podemos encontrar algunas características definidas en la Tabla 5:

Tabla 5 Resumen de comparación entre el análisis hecho para DBSCAN y para HDBSCAN para los distintos meses

Aspecto	**Análisis DBSCAN**	**Análisis HDBSCAN**
Abril	Grupo con alta conectividad	Aislado como clúster intermedio
Mayo	Grupo de baja conectividad	Mismo grupo con Junio y Septiembre
Junio y Septiembre	Agrupados con Abril (alta conectividad)	Agrupados con Mayo (muy baja conectividad)
Julio	En grupo de baja conectividad	En grupo con mayor tamaño medio de nodo
Agosto	Aislado como clúster por su densidad	Agrupado con Julio

Vemos, por tanto, que las diferencias fundamentales entre ambos algoritmos de *clustering* tienen que ver con el número de nodos de los grafos y también con el número de aeropuertos en cada nodo. En este sentido, HDBSCAN no muestra nodos con un solo aeropuerto, con lo que estos grafos resumen de forma más general las características de la red aeroportuaria en esos meses, agrupando solamente los aeropuertos con similitudes. Vemos que genera pocos nodos con muy pocas conexiones. DBSCAN ofrece un detalle más granular en la representación al permitir aeropuertos aislados y ofrece alguna estructura en tronco más interesante con más uniones entre nodos.

Como se comenta en el apartado de conclusiones, las interpretaciones posibles de estos grafos están limitadas por la poca información que se utiliza: en realidad, solo se usan dos piezas de información, la desviación entre trayectorias y el retraso del vuelo. Por ello, los aeropuertos están agrupados en base a estas dos características y, aunque puedan estar relacionadas con la densidad del tráfico aéreo en un aeropuerto, no lo están midiendo directamente. Por ello, no hay una base teórica firme por la que estas agrupaciones debieran coincidir con las de AENA.

4 Conclusiones

En este capítulo se recogen las conclusiones obtenidas y se comentan las limitaciones que tiene este proyecto y en qué afectan a los resultados obtenidos, así como los trabajos futuros que se plantean para ampliar el trabajo realizado.

4.1 Conclusiones

El objetivo principal, que consistía en la transformación del conjunto desestructurado inicial de datos sobre trayectorias de vuelos (reales y planificadas) con salida y llegada los 46 aeropuertos españoles en un conjunto estructurado de variables listas para ser incorporadas en cualquier proceso genérico de análisis de datos, se ha conseguido con éxito. Los ficheros iniciales, con información sobre 1134813 vuelos registrados entre el 25/03/2018 y el 28/10/2018 entre los aeropuertos españoles se sometieron a un proceso detallado de depuración, en los que se eliminaron los vuelos circulares, duplicados, cancelados, desviados o con inconsistencias temporales, obteniendo un total de 1098572 vuelos, eliminando un 3,2% de los vuelos originales. Se ha logrado extraer de forma minuciosa la información contenida en las trayectorias real y planificada, sincronizándolas por normalización del progreso espacial, expresándolas en la misma escala para que sean comparables. A partir de ahí, una cuidadosa interpolación en cada ruta ha permitido calcular la distancia punto a punto entre ambas y, utilizando la distancia de Haversine, ha sido posible reducir cada par de trayectorias a un único valor que nos informa de la distancia entre las mismas. Esta información se ha complementado con la diferencia de altitudes del avión entre ambas trayectorias y el retraso (o adelanto) del vuelo al alcanzar el destino programado. El conjunto de datos está listo para ser utilizado como entrada en otros procesos de análisis de datos.

Un objetivo secundario, pero no menos importante, era el uso del conjunto de datos transformado y limpio. A través de la aplicación del algoritmo Mapper se consigue proyectar un espacio de muy alta dimensión en un grafo simplificado que trata de mantener la conectividad de los datos. Para ello, se calculan las características descriptivas media, mediana, desviación típica y rango intercuartílico de las variables distancia entre trayectorias y retraso, para los vuelos agrupados por aeropuertos. Se han probado distintas agregaciones de los datos y distintas configuraciones de parámetros, así como dos algoritmos de *clustering*, extrayéndose ideas interesantes en cada una de ellas. Las agrupaciones obtenidas y sus conexiones no siempre coinciden con la clasificación de aeropuertos realizada por AENA, que se basa exclusivamente en la densidad anual del tránsito de pasajeros del aeropuerto. En las agrupaciones mostradas en este proyecto, los aeropuertos similares lo son en términos de distancias entre las trayectorias reales y planificadas de sus vuelos, así como en los retrasos experimentados. Estas características no tienen por qué estar relacionadas directamente con el tránsito de pasajeros ya que dependen de una combinación de factores técnicos, operativos, meteorológicos y regulatorios [22]. Todas estas variables pueden estar recogidas de manera indirecta en las distancias calculadas entre trayectoria real y planificada y no están consideradas de ninguna forma en la clasificación hecha por AENA. De ahí que las clasificaciones aquí obtenidas y la hecha por AENA no tengan por qué coincidir y si lo hacen sea de manera anecdótica.

En resumen, el proyecto presenta una aplicación del Análisis Topológico de Datos, y en particular del algoritmo Mapper, en el análisis de trayectorias de vuelos, para comprender la complejidad del espacio aéreo español, ofreciendo representaciones alternativas a las ya existentes.

4.2 Limitaciones

Este proyecto presenta diversas limitaciones que es importante tener en consideración. En primer lugar, los datos se centran en la temporada de verano de 2018. Esto es únicamente una época de un año concreto, por lo que los patrones encontrados pueden variar en otras ventanas de tiempo y los resultados pueden ser aún más distintos si se comparan con otros años como puede ser el año de la pandemia.

Una limitación de gran consideración es que las características que se recogen finalmente de los vuelos solo son dos: el retraso y la desviación entre la ruta planificada y la real. Al no conocer la naturaleza y características específicas de los vuelos, se tratan todos por igual para los distintos aeropuertos lo que puede sesgar la comparación entre los mismos.

Además, no se está teniendo en cuenta una gran consideración de variables como pueden ser las variables meteorológicas, que aportan una información crucial acerca de las razones por las que los vuelos sufren de unos retrasos o desviaciones.

4.3 Trabajos futuros

El proyecto ha conseguido con éxito demostrar la gran aplicabilidad que tiene el Análisis Topológico de Datos para analizar las trayectorias de las aeronaves de los aeropuertos españoles. Algunas de las líneas de trabajo futuras que permitirían extender y mejorar los resultados aquí obtenidos son:

- Para ver la evolución temporal de los clústeres de aeropuertos, se pueden usar datos desagregados por fecha, por día de operación, incluyendo quincenas o meses en el conjunto de datos de entrada del algoritmo. Así se puede estudiar la evolución temporal de la misma manera que en [23].
- Se pueden estudiar los mapas de salidas y llegadas por separado, para ver si difieren mucho de los obtenidos juntando todas las variables. Por ejemplo, en un mapa de solo llegadas, los clústeres agruparían aeropuertos similares por condiciones específicas en la llegada de vuelos (climatológicas, geográficas, de ruta, etc.)
- Una exploración más profunda del sistema de extracción de datos NEST para la identificación de nuevas variables que pudieran ser útiles en este tipo de análisis, por ejemplo, compañía que opera el vuelo, tipo de avión utilizado, condiciones meteorológicas en la ruta, condiciones del espacio aéreo como la congestión, etc. En concreto, analizar las distancias entre trayectorias

dependiendo del tipo de avión puede mejorar la interpretabilidad de los resultados mostrados en los gráficos.

- Estudiar la manera óptima de incorporar la información proporcionada por las nuevas variables en el conjunto de datos final.
- Explorar nuevas representaciones en Mapper usando otros filtros (*lens*) como t-SNE.
- Extender el horizonte temporal. Incorporar una mayor cantidad de datos, cubriendo un mayor horizonte temporal. Por ejemplo, 10 años de datos, permitiría analizar la evolución del tráfico aéreo. Además, permitiría estudiar la capacidad del Análisis Topológico de Datos de analizar datos que sufren algún tipo de cambio drástico o shock como puede ser la pandemia del COVID-19 o la crisis de las guerras.
- Integrar la homología persistente: Integrar, además, la homología persistente y los paisajes de persistencia junto al algoritmo Mapper permitiría, no solo analizar las similitudes y diferencias entre ambas ramas del Análisis Topológico, si no también cuantificar formalmente los vacíos y ciclos que hay en las trayectorias.
- Creación de una herramienta en tiempo real. Si se pudiera migrar la herramienta para procesar planes de vuelo en *streaming* de forma casi instantánea se podrían generar alertas topológicas al supervisor de la red aérea.

5 Referencias

[1] L. Varley, «Global Aviation Sets a Passenger Seat Record in 2024,» 20 01 2025. [En línea]. Available: https://aviationsourcenews.com/global-aviation-sets-a-passenger-seat-record-in-2024/.

[2] IATA, «Global Air Passenger Demand Reaches Record High in 2024,» 30 01 2025. [En línea]. Available: https://www.iata.org/en/pressroom/2025-releases/2025-01-30-01/.

[3] J. H. M. Luigi Raphael I. Dy, «Validating ADS-B Data for Use in Noise Modeling Applications,» *University Aviation Association (UAA),* vol. 40, nº 2, p. 14, 2022.

[4] L. G. R. M. A. V. V. F. G. C. Manuel Cuerno, «TOPOLOGICAL DATA ANALYSIS IN ATM: THE SHAPE OF BIG FLIGHT,» p. 31, 2023.

[5] B. M. Frédéric Chazal, «An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists,» 29 09 2021. [En línea]. Available: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2021.667963/full.

[6] M. S. R. H. B. Max Z. Li, «Topological data analysis for aviation applications,» *Transportation Research Part E,* p. 26, 2019.

[7] A. F. Miriam Esteve, «Trajectory Classification through Topological Data Analysis Perspectives,» *Department of Matematicas, Fısica y Ciencias Tecnologicas, Universidad,* p. 22, 2024.

[8] CRIDA, «Centro de Referencia de Investigación,,» [En línea]. Available: https://crida.es/webcrida/.

[9] Eurocontrol, «Research network strategy monitoring tool,» Eurocontrol, [En línea]. Available: https://www.eurocontrol.int/solution/rnest.

[10] Python, «python,» [En línea]. Available: https://www.python.org/.

[11] OACI, «OACI Special Designators,» OACI, [En línea]. Available: https://www.icao.int/publications/DOC8643/Pages/SpecialDesignators.aspx. [Último acceso: 17 06 2025].

[12] «Aeropuertos nacionales e internacionales,» 27 02 2011. [En línea]. Available: https://aeropuertos.wordpress.com/2011/02/27/listado-aeropuertos-espana/. [Último acceso: 02 04 2025].

[13] C. Veness, «Calculate distance, bearing and more between Latitude/Longitude points,» Movable Type Scripts, [En línea]. Available: https://www.movable-type.co.uk/scripts/latlong.html.

[14] S. Talebi, «Topological Data Analysis (TDA),» Medium, 21 05 2022. [En línea]. Available: https://medium.com/data-science/topological-data-analysis-tda-b7f9b770c951.

[15] S. Talebi, «Persistent Homology,» Medium, 16 06 2022. [En línea]. Available: https://medium.datadriveninvestor.com/persistent-homology-f22789d753c4.

[16] G. a. M. F. a. C. G. Singh, «Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition,» Eurographics Symposium on Point-Based Graphics, 2007. [En línea]. Available: https://research.math.osu.edu/tgda/mapperPBG.pdf.

[17] S. Talebi, «The Mapper Algorithm,» Medium, 03 06 2022. [En línea]. Available: https://medium.datadriveninvestor.com/the-mapper-algorithm-d0842f926658.

[18] B. C. U. N. F. S. Z. Vine Nwabuisi Madukpe, «A Comprehensive Review of the Mapper Algorithm, a Topological Data,» p. 31, 2025.

[19] S. Navarro, «¿Qué es el Hierarchical DBSCAN (HDBSCAN)?,» 7 11 2024. [En línea]. Available: https://keepcoding.io/blog/que-es-hierarchical-dbscan-o-hdbscan/.

[20] N. S. D. E. S. M. Hendrik Jacob van Veen, «KeplerMapper 2.1.0 documentation,» scikit-tda, [En línea]. Available: https://kepler-mapper.scikit-tda.org/en/latest/.

[21] AENA, «Estadísticas de tráfico aéreo,» 2025. [En línea]. Available: https://www.aena.es/es/estadisticas/inicio.html.

[22] «Flight planning,» Wikipedia, 27 06 2025. [En línea]. Available: https://en.wikipedia.org/wiki/Flight_planning.

[23] I. A. Mendoza, «Análisis topológico de datos espacial de Covid 19 en España,» *Tésis de Máster. Escuela Técnica Superior de Ingenieros Informáticos,* 2024.

Anexos

Anexo 1: Grafos

A continuación, se muestran las figuras de los mapas obtenidos para los meses de abril, mayo, junio, julio y setiembre a los que hace referencia la Sección 3.2.3

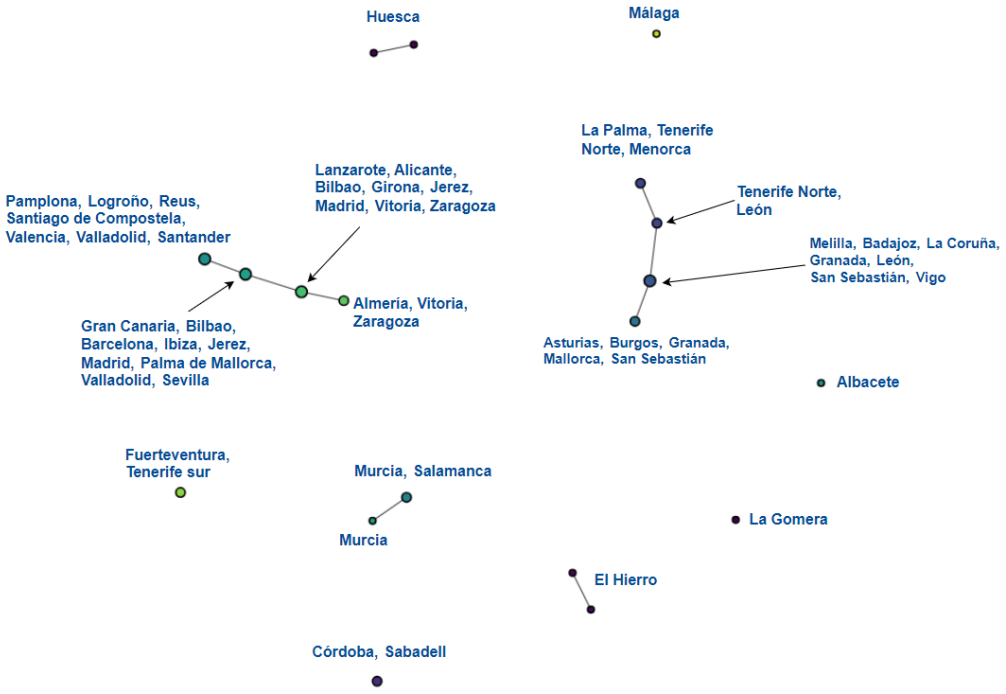

Figura 0.1 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando DBSCAN, de los vuelos de abril

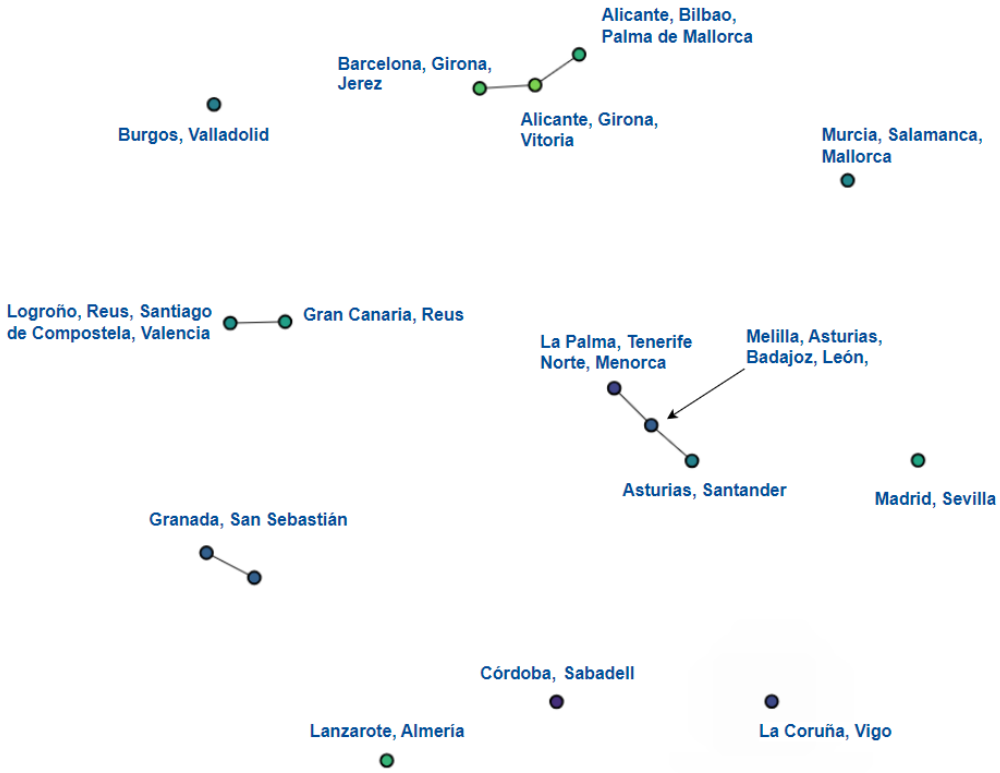

Figura 0.2 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando HDBSCAN, de los vuelos de abril

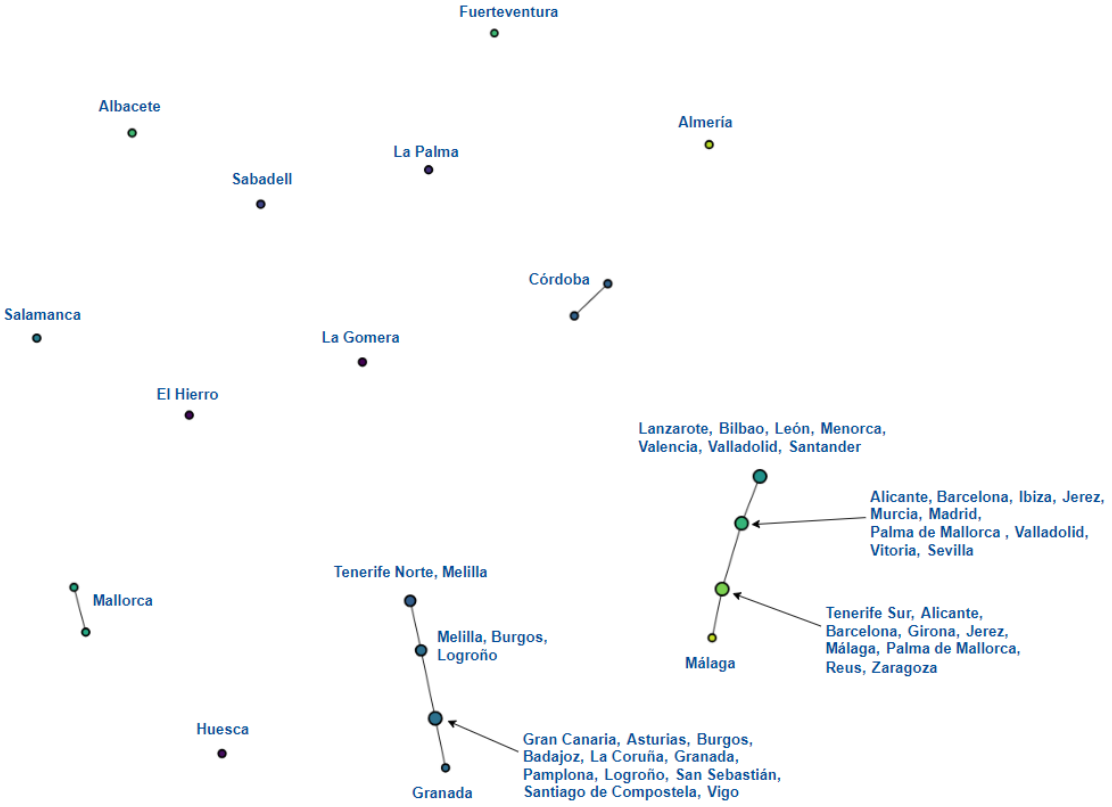

Figura 0.3 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando DBSCAN, de los vuelos de mayo

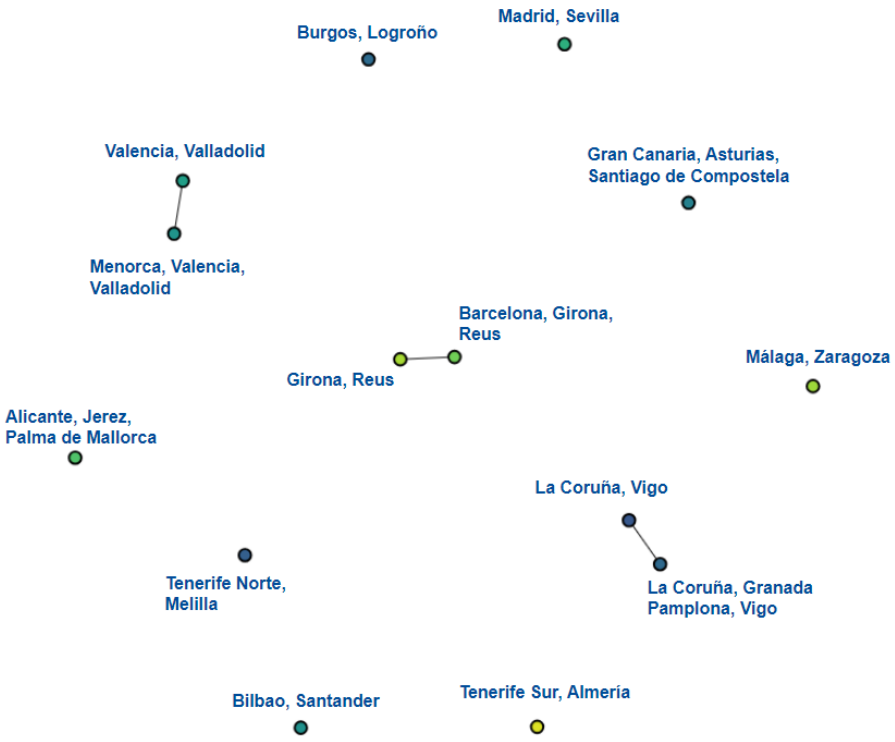

Figura 0.4 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando HDBSCAN, de los vuelos de mayo

Figura 0.5 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando DBSCAN, de los vuelos de junio

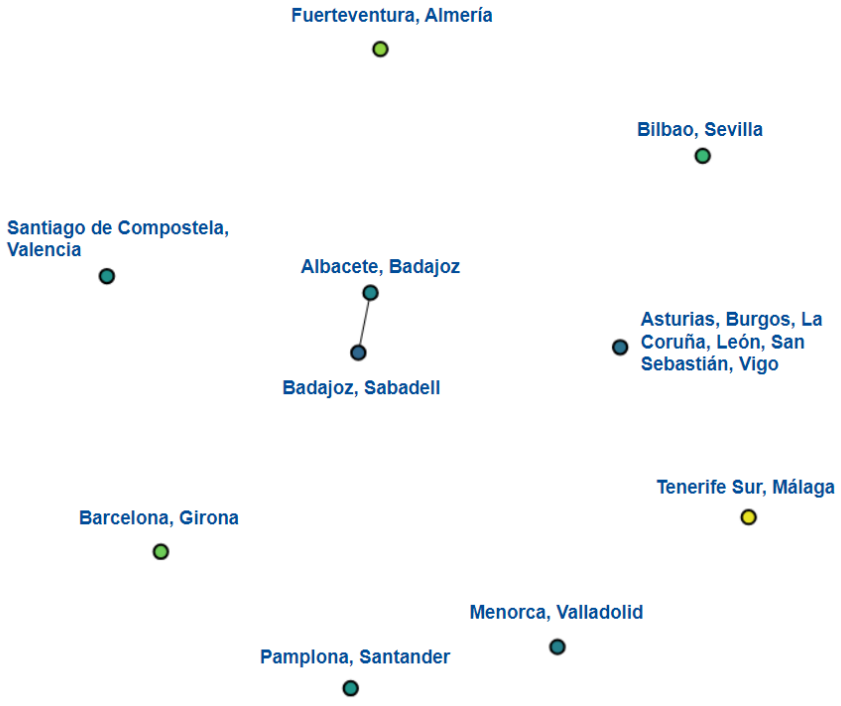

Figura 0.6 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando HDBSCAN, de los vuelos de junio

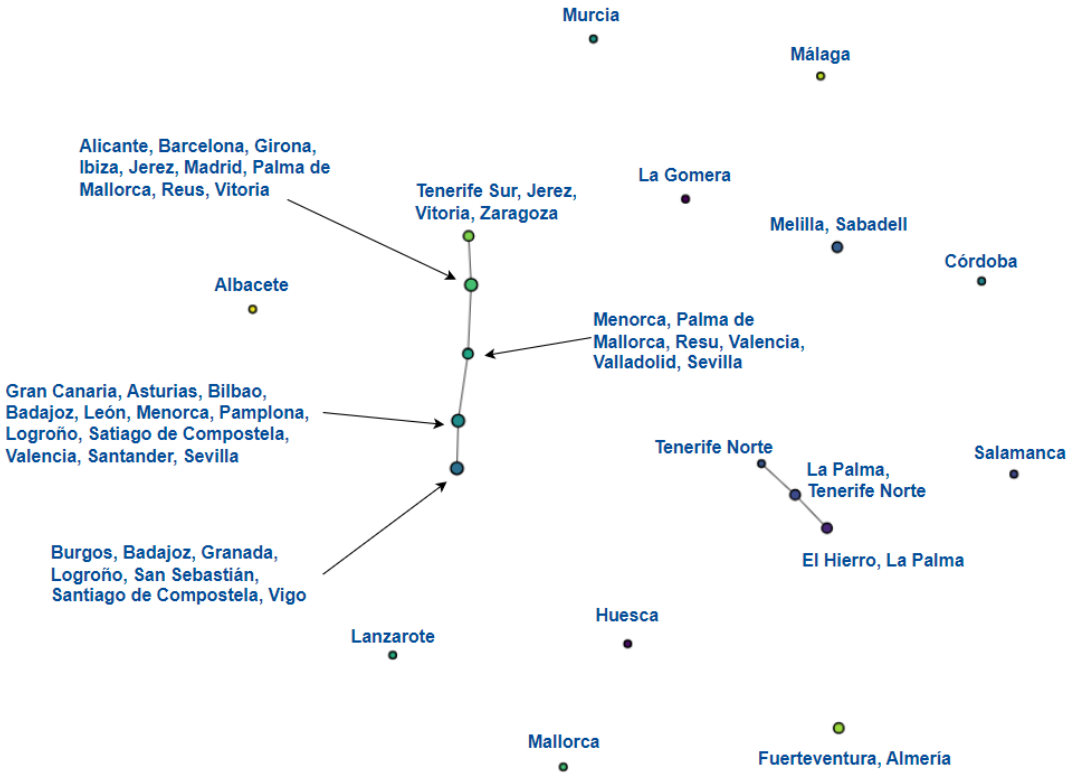

Figura 0.7 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando DBSCAN, de los vuelos de julio

Figura 0.8 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando HDBSCAN, de los vuelos de julio

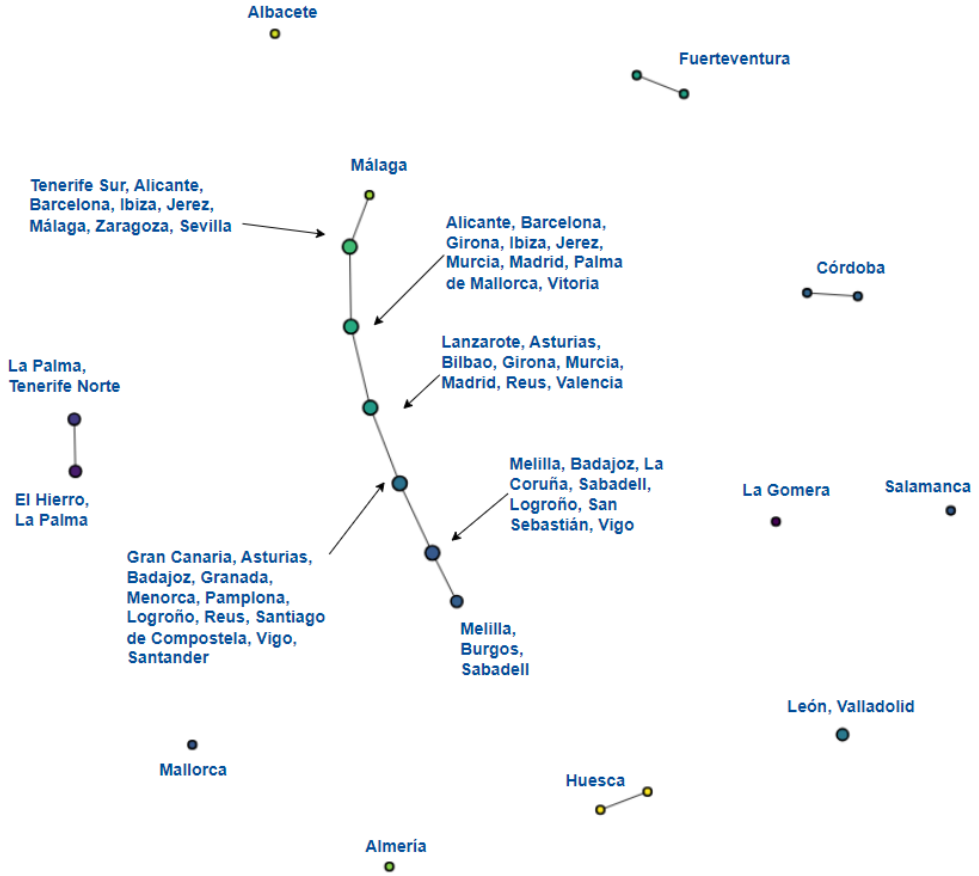

Figura 0.9 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando DBSCAN, de los vuelos de septiembre

Figura 0.10 Grafo resultante del algoritmo Mapper con vuelos agrupados por aeropuertos, utilizando HDBSCAN, de los vuelos de septiembre

Anexo 2: Repositorio en Github del proyecto

https://github.com/raulserranoc4/Mapper-TFM/tree/main

Anexo 3: Carpeta compartida con los gráficos dinámicos del Mapper

https://drive.google.com/drive/u/0/folders/1m-chEzcoFLfj6K1jaBnxgQkOQEFA8exF