Universidad Zaragoza
1542

Trabajo Fin de Máster

Clasificación de estilos de conducción utilizando técnicas de machine learning

Classification of driving behaviour using machine learning techniques

Autor

Luis Llorente Muro

Director

Julio David Buldain Pérez

Departamento de Ingeniería Electrónica y Comunicaciones

Escuela de Ingeniería y Arquitectura

2024-2025

En este futuro brillante no puedes olvidar tu pasado.

BOB MARLEY,

Cantante y compositor jamaicano.

A mis padres y amigos, a Buldain y a Centro Zaragoza.

Muchas gracias.

RESUMEN

Este trabajo se realiza en el marco de un proyecto de I+D de la empresa Centro Zaragoza, enfocado en la caracterización de estilos de conducción. Se utilizan datos recopilados de 16,851 viajes realizados por 107 usuarios.

Tras un meticuloso proceso de filtrado y de ingeniería de características que aseguren la calidad y precisión del análisis, se han aplicado técnicas de aprendizaje automático que sugieren 2 ó 4 clusters, principalmente. Estas agrupaciones se pueden sintetizar en conductores prudentes e imprudentes, que a su vez pueden ser clasificados como agresivos, temerarios y distraídos, basándose en tres características principales: tiempo excediendo la velocidad permitida, aceleraciones bruscas y uso del móvil.

Los resultados obtenidos abren la puerta a numerosas aplicaciones en distintos sectores que pueden utilizar estos servicios para personalizar las primas de seguros o planificar una movilidad mucho más eficiente y sostenible, entre otros.

ABSTRACT

This work is carried out within the framework of an R&D project by Centro Zaragoza, focusing on the characterization of driving styles. The study utilizes data collected from 16,851 trips made by 107 users.

After a meticulous process of data filtering and feature engineering to ensure the quality and accuracy of the analysis, machine learning techniques were applied, suggesting primarily 2 or 4 clusters. These clusters can be summarized as cautious and reckless drivers, further classified into aggressive, reckless, and distracted categories based on three main characteristics: time spent exceeding the speed limit, sudden accelerations, and mobile phone usage.

The results obtained pave the way for numerous applications across different sectors that can use these insights to personalize insurance premiums or plan for more efficient and sustainable mobility, among other uses.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

1.1. Motivación

El presente trabajo se lleva a cabo en el marco de un proyecto de I+D en curso en mi actual empresa, dedicado específicamente a la caracterización de estilos de conducción. Se ha desarrollado en las instalaciones de Centro Zaragoza, un instituto de investigación destacado en el sector automovilístico por su compromiso con la innovación y el desarrollo tecnológico.

En el primer capítulo se introduce el problema a resolver y los objetivos pretendidos, así como una breve descripción del estado del arte. En el segundo, se presenta una descripción detallada del conjunto de datos que se tiene, además del proceso de limpieza y procesado del mismo. En el tercer capítulo se presentan los resultados obtenidos de los que se extraen una serie de conclusiones que aparecen descritas en el último apartado.

La motivación detrás de este trabajo radica en la creciente importancia de adaptar los sistemas de asistencia al conductor y las tecnologías de seguridad vehicular a las necesidades específicas y estilos de conducción de los usuarios. En un mundo donde la movilidad se enfrenta a desafíos sin precedentes relacionados con la seguridad, la eficiencia energética y la sostenibilidad, comprender los diferentes estilos de conducción se convierte en un requisito esencial para el diseño de vehículos más seguros, eficientes y personalizados.

Además, este proyecto se alinea con la necesidad global de avanzar hacia soluciones de movilidad más sostenibles. Al optimizar la conducción a través de la personalización tecnológica, se espera contribuir a la reducción de emisiones contaminantes y al fomento de prácticas de conducción más eficientes y responsables.

Este trabajo representa, por lo tanto, un esfuerzo por vincular la investigación académica con las necesidades prácticas del sector automotriz, ofreciendo una oportunidad única para contribuir a la evolución de la movilidad del futuro desde una perspectiva innovadora y sostenible.

1.2. Descripción del problema

La conducción es una actividad compleja que involucra una combinación de habilidades cognitivas, psicomotoras y perceptuales. Cada conductor posee un estilo único, influenciado por su experiencia, personalidad y contexto cultural, entre otros factores. La caracterización efectiva de estos estilos de conducción abre la puerta a una multitud de aplicaciones benéficas: desde sistemas de vehículos autónomos que se adaptan al estilo personal de conducción hasta programas de aseguradoras que personalizan las primas según el riesgo real que cada estilo conlleva.

Se trabajará con una muestra facilitada por Centro Zaragoza, que contiene 16851 viajes de 107 usuarios diferentes que suman un total de 359574.61 kilómetros recorridos y 1377344.8 horas.

Los viajes provienen de una aplicación móvil desarrollada por una empresa externa que colabora con Centro Zaragoza en distintos proyectos. Esta herramienta recopila información sobre los viajes realizados por usuarios que voluntariamente participan en el estudio. Durante el trayecto, la aplicación utiliza el GPS del dispositivo móvil del usuario para monitorizar y almacenar variables dinámicas del viaje.

Se cuenta con dos *datasets* principales; *dataset* de viajes y *dataset* de posiciones:

El primero contiene información de un trayecto completo registrado por la aplicación, indicando fecha y hora de inicio del viaje, el tipo de vehículo o la distancia recorrida en kilómetros, entre otros parámetros. Por otro lado, el *dataset* de posiciones contiene información relevante como la posición GPS, la velocidad en Km/h o la velocidad máxima permitida en la vía.

1.3. Objetivos

Como se viene comentando, el objetivo principal de este trabajo es caracterizar estilos de conducción. Para ello, se seguirá una metodología muy definida estructurada en cuatro fases principales:

1.3.1. Estudiar bibliografía sobre técnicas de clasificación de tipos de conducción

Esta primera fase proporcionará una sólida base teórica sobre las técnicas de clasificación no supervisada aplicadas a la identificación de estilos de conducción. Se realizará una revisión exhaustiva de la literatura científica y técnica más relevante para comprender los métodos actuales y las tendencias en este campo de investigación. Las fuentes principales incluirán bases de datos académicas como IEEE Xplore, Google Scholar y ResearchGate.

1.3.2. Limpiar y procesar los datos proporcionados

Tras el estudio detallado de la bibliografía, se analizarán preliminarmente los distintos *datasets* proporcionados que contienen datos de conducción recogidos de distintos vehículos. La finalidad es comprender la estructura de los datos, la calidad y la idoneidad para su uso en la clasificación de estilos de conducción mediante técnicas no supervisadas. Este análisis incluirá los siguientes puntos clave:

a) **Carga y lectura de datos:** Se importarán las librerías necesarias para la conversión de los distintos ficheros a un *DataFrame*.

b) **Análisis de los *datasets*:** Se estudiarán las distintas variables que contienen los ficheros proporcionados y se realizará un primer análisis cuantitativo de los datos.

c) Preparación de los datos: Se unificarán todos los ficheros en un único *dataset* y se realizarán una serie de procedimientos de limpieza antes de utilizarlo para el análisis de la conducción.

d) *Feature engineering*: Se llevará a cabo un proceso para seleccionar y transformar las variables más relevantes para la identificación de patrones de conducción.

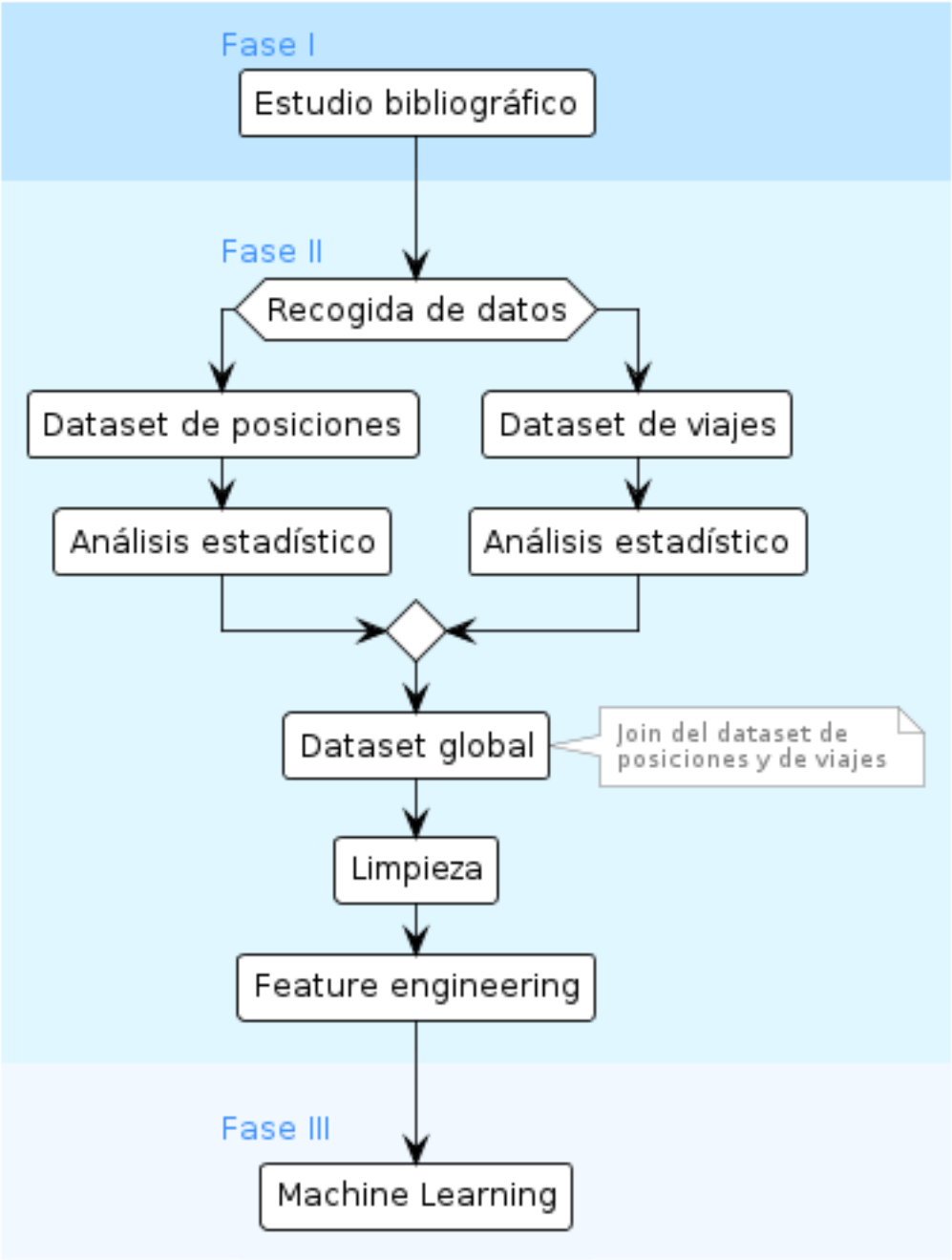


Figura 1. Proceso de preparación de los datos.

1.3.3. Implementar técnicas de aprendizaje automático

En esta fase, se explorarán y aplicarán diversas técnicas de aprendizaje automático utilizadas en problemas no supervisados. Se probarán tanto métodos populares y sencillos de implementar como otros más avanzados y complejos. Estas técnicas incluirán métodos de clustering que varían en complejidad y enfoque para evaluar cuál se adapta mejor a los datos en cuestión.

1.3.4. Analizar los resultados obtenidos

Finalmente, se evaluarán los distintos algoritmos para determinar cuál ofrece la mejor interpretación y clasificación de los estilos de conducción para los datos de los que se dispone.

1.4. Planificación

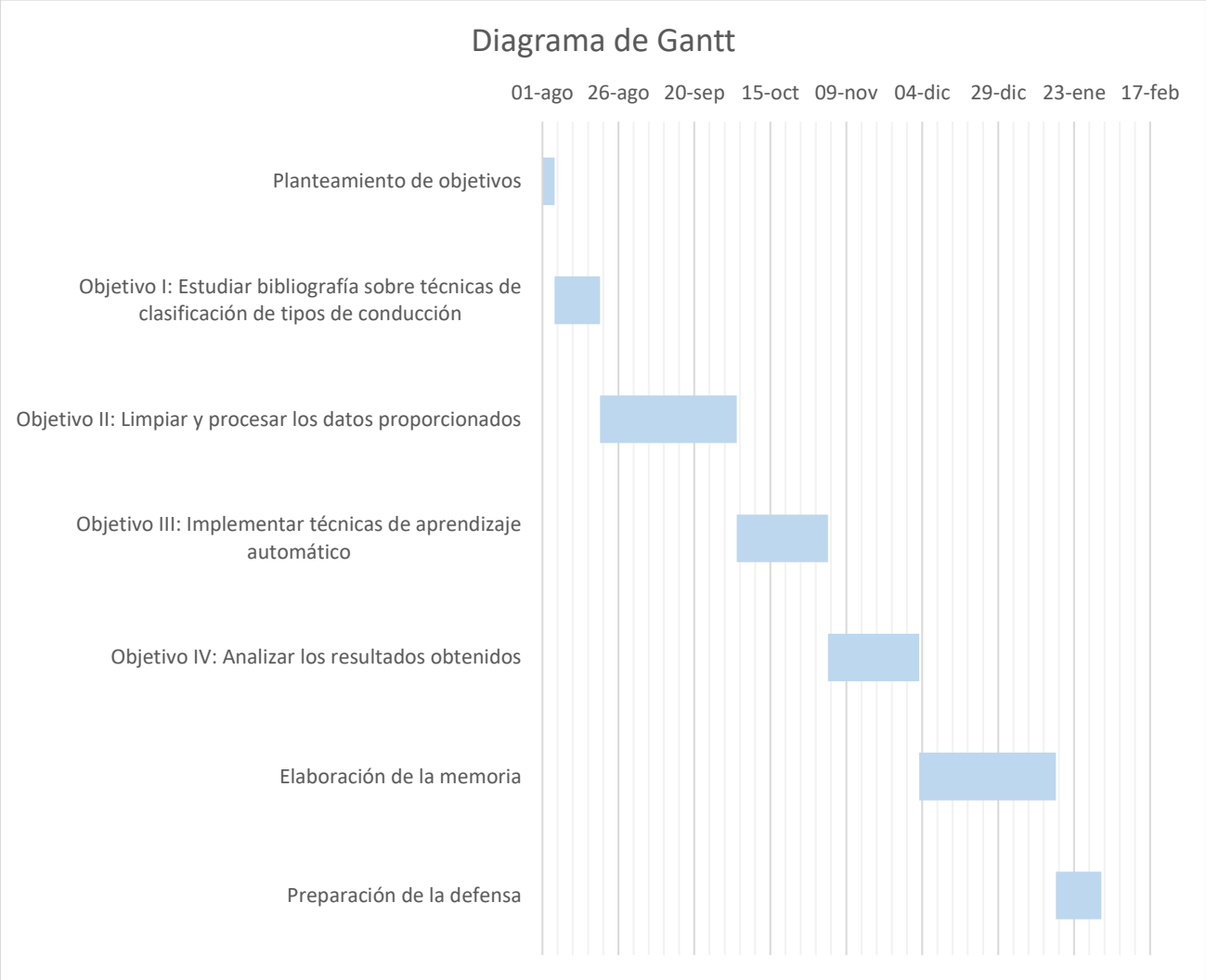


Figura 2. Planificación de este trabajo.

1.5. Material empleado

El desafío reside en cómo capturar, analizar e interpretar los vastos conjuntos de datos generados durante los viajes. Aquí es donde entra en juego el poder de la ciencia de datos y las herramientas tecnológicas avanzadas. Este trabajo emplea un conjunto sofisticado de herramientas para el manejo, procesamiento y análisis de grandes volúmenes de datos, cada una seleccionada por sus capacidades específicas y su sinergia con las demás.

Python: Conocido por su sintaxis clara y su poderosa biblioteca de análisis de datos, Python se ha establecido como el lenguaje predilecto en el campo de la ciencia de datos. Su versatilidad permite desde la manipulación básica de datos hasta la implementación de complejos algoritmos de aprendizaje automático, convirtiéndolo en una herramienta indispensable para este proyecto [1]. La amplia comunidad de usuarios y desarrolladores de Python asegura un soporte continuo y una rica oferta de recursos educativos, lo que facilita la solución de problemas y la implementación de nuevas técnicas.

Anaconda: Para gestionar eficientemente las dependencias de Python y facilitar la reproducibilidad del análisis, se ha utilizado Anaconda. Esta plataforma libre y abierta simplifica la gestión de paquetes y entornos de trabajo en Python, permitiendo una configuración rápida y sencilla del entorno de desarrollo necesario para el proyecto [2]. Su selección se debe a la capacidad de Anaconda para resolver complejidades asociadas con la instalación de paquetes y sus dependencias, ahorrando tiempo y evitando potenciales conflictos entre librerías.

Spyder y Jupyter Notebook: En cuanto a los entornos de desarrollo, se ha optado por Spyder y Jupyter Notebook. Spyder ofrece un entorno de desarrollo integrado (IDE) que es amigable para quienes vienen de otros lenguajes de programación, con características que facilitan la depuración y el análisis de datos [3]. Por su parte, Jupyter Notebook se destaca por su capacidad de combinar código, visualizaciones y texto en un solo documento, lo que lo hace ideal para la exploración de datos y la presentación de resultados preliminares [4]. La elección de estas herramientas se basa en su capacidad para adaptarse a diferentes etapas del análisis de datos, desde la exploración inicial hasta la presentación final de los resultados.

Control de versiones con Git: Finalmente, el control de versiones se ha gestionado mediante Git, un sistema diseñado para rastrear cambios en los archivos. En un proyecto de investigación donde el código evoluciona constantemente, Git permite registrar las modificaciones y revertir a versiones anteriores si es necesario, garantizando así la integridad del trabajo y permitiendo un desarrollo iterativo del proyecto [5].

La combinación de estos recursos tecnológicos con metodologías avanzadas de análisis de datos posibilita una exploración profunda de los estilos de conducción, superando las limitaciones de los estudios tradicionales basados en observaciones directas o encuestas. El enfoque basado en datos permite identificar patrones complejos y sutiles en el comportamiento de conducción que de otro modo serían imperceptibles.

Comprender los estilos de conducción no solo tiene implicaciones en la personalización y seguridad de los sistemas de asistencia al conductor, sino que también contribuye significativamente a la prevención de accidentes, la planificación urbana y la reducción del impacto ambiental del transporte. Al ajustar los sistemas de vehículos para complementar o corregir ciertas tendencias de conducción, se puede mejorar la seguridad de todos los usuarios de la vía. Además, los datos sobre estilos de conducción pueden informar el diseño de políticas de tráfico y movilidad urbana más efectivas, promoviendo un uso más seguro y eficiente de las infraestructuras viales.

Este trabajo se sitúa, por tanto, en la intersección de la tecnología y la seguridad vial, aportando no solo al campo académico mediante la aplicación de técnicas avanzadas de análisis de datos, sino también ofreciendo enfoques prácticos que pueden ser implementados por fabricantes de automóviles, urbanistas y responsables de la formulación de políticas de tráfico y seguridad. La ambición es avanzar hacia un futuro donde la movilidad sea más segura, eficiente y sostenible.

1.6. Estado del arte

Una vez obtenidos los datos, se hace necesaria la ejecución de una serie de procedimientos de limpieza antes de utilizarlos para el análisis de la conducción. Estos procedimientos suelen incluir la exclusión del ruido y *feature engineering*.

Gestionar el ruido es crucial, especialmente en las mediciones de acelerómetros que se ven perturbadas por vibraciones o baches debido a la gravedad [6], [7].

La precisión horizontal de las mediciones GPS, que se proporciona junto a otros valores, ha sido empleada para depurar datos con ruido [8], [9]. Otros enfoques más avanzados incluyen el uso de redes neuronales con retardo de entrada (IDNN) [10] o filtros de Kalman [11]. En diversos estudios [12], [13] se implementan filtros de paso bajo o medias móviles simples para mitigar el ruido.

En el ámbito de la extracción de características específicas, la mayoría de los estudios dedicados al análisis de la conducción han diseñado sistemas para identificar eventos de aceleración y frenado bruscos. En [8], los investigadores proponen un algoritmo de fusión que distingue entre aceleración normal y aceleración o frenado bruscos en un ciclo de conducción. Este enfoque integra datos de GPS,

acelerómetro, posición del vehículo y diferencias temporales entre puntos consecutivos del recorrido. Por su parte, en [14] los investigadores desarrollan un clasificador de aprendizaje automático para detectar maniobras anómalas, como frenadas bruscas, cambios repentinos de carril, curvas riesgosas, excesos de velocidad y desviaciones en la ruta.

Otra característica de interés es el uso del teléfono móvil mientras se conduce. Esta información principalmente se emplea para identificar si el conductor está distraído de la tarea de conducción real [15], [16]. La detección de actividades secundarias, como enviar mensajes de texto o hablar por teléfono mientras se maneja, no solo ofrece una mejor comprensión del comportamiento y la distracción del conductor, sino que también ayuda a eliminar el ruido originado por los movimientos bruscos del teléfono. En [17], los investigadores han determinado si el conductor estaba realizando acciones como enviar mensajes de texto, llamar o leer mientras conducía.

Finalmente, la investigación actual está enfocada en identificar prácticas de conducción ecológicas que promuevan una menor utilización de combustible. La conducción ecológica puede ser identificada a través de indicadores que describen las condiciones del vehículo y del tráfico, tales como los pares de revoluciones del motor y la marcha, junto con los cambios en la energía cinética del vehículo [18].

En ausencia de estos datos, las técnicas de minería de datos se aplican exclusivamente a medidas de velocidad y aceleración para detectar patrones de conducción que fomenten un consumo de combustible eficiente. Las principales características extraídas para evaluar si el conductor adopta un enfoque de ecoconducción incluyen los índices de velocidad, aceleración y dirección [19].

Aunque una gran cantidad de literatura utiliza métodos de análisis estadístico para investigar varios comportamientos de conducción, en las últimas décadas los enfoques de aprendizaje automático han ganado terreno en este campo. En [20] se han utilizado técnicas de aprendizaje automático para identificar eventos de baches y frenazos demostrando la importancia de las técnicas de filtrado y aprendizaje automático (agrupación de k-Means y Support Vector Machines - SVM) para una mejor identificación de eventos en la conducción. En [21] se adoptan la lógica difusa y un algoritmo basado en reglas jerárquicas para detectar comportamientos de conducción, como aceleración y frenado bruscos o dirección agresiva, utilizando los datos del acelerómetro y el giroscopio. En [22], los investigadores utilizaron el modelo de mezclas gaussianas (GMM) con el método de periodograma para clasificar el comportamiento de conducción en un gradiente de comportamiento suave a agresivo.

Otros investigadores han propuesto reconocer las características del conductor mediante la implementación de algoritmos de clasificación. En concreto, en [23] se compararon varios algoritmos de clasificación con respecto a su rendimiento en la identificación de tres estados de conducción

distintos: normal, somnoliento y agresivo. Los resultados indicaron que Random Forest tuvo la mayor precisión general en comparación con los otros clasificadores (k-Nearest Neighbors, Decision Tree, SVM). En [24], el algoritmo MODLEM logró la máxima precisión en comparación con otros algoritmos de clasificación para detectar eventos de frenado utilizando datos del acelerómetro. En [15] se desarrolló un algoritmo de agrupación de k-Means en dos pasos para distinguir inicialmente los viajes agresivos de los no agresivos y, a continuación, estos viajes ya clasificados se volvieron a clasificar respecto a la distracción del conductor y la asunción de riesgos.

Por su parte, las redes neuronales han sido utilizadas en [25] para identificar el grado de agresividad del conductor utilizando mediciones de velocidad y aceleración. [26] evaluaron el rendimiento de dos algoritmos de clasificación (Árbol de decisión y Naïve Bayes) en comparación con el de una red neuronal, sus hallazgos revelaron que la red neuronal supera a los otros métodos en la detección de maniobras de conducción. Sin embargo, como la potencia de cálculo necesaria para el entrenamiento y la validación de los modelos aumenta considerablemente, es necesario realizar varias tareas offline.

Con la disponibilidad de conjuntos de datos masivos de teléfonos inteligentes, se vio que, si bien los métodos estadísticos podían proporcionar una primera visión de los conjuntos de datos, se necesitaban técnicas más avanzadas para diseñar soluciones precisas y eficientes a los diferentes retos. Un ejemplo de ello puede encontrarse en [14], que desarrollaron clasificadores avanzados para detectar patrones de conducción bruscos y obtuvieron mejores resultados en comparación con los métodos clásicos de análisis de actividad a partir de datos de acelerómetro basados en métricas estadísticas de desviación estándar, entropía, energía, valor medio, etc.

Además, como se muestra en la literatura, los investigadores suelen utilizar métodos estadísticos cuando se detectan eventos aislados o se separa el comportamiento de conducción anormal del de conducción segura. Por el contrario, los métodos basados en Machine Learning se utilizan cuando se investiga toda la gama de comportamientos de conducción y se detectan varios perfiles de conducción diferentes. En [27] se comparan diferentes enfoques en términos de precisión de la clasificación.

2. MATERIALES Y MÉTODOS

2.1. Descripción previa de los datos

Tal y como se viene comentando, se trabaja con una muestra que contiene 17351 viajes de 107 usuarios diferentes que suman un total de 359574.61 kilómetros recorridos y 1377344.8 horas. Se cuenta con dos *datasets* principales; *dataset* de viajes y *dataset* de posiciones.

2.1.1. *Dataset* de viajes

Este *dataset* contiene información de un trayecto completo. Cada columna proporciona distinta información relevante sobre los viajes:

- **date:** Indica la fecha y hora en que comenzó el viaje, en formato *datetime*.
- **initialTime:** La hora de inicio del viaje, expresada en milisegundos desde el 1 de enero de 1970, en formato numérico.
- **endTime:** Hora en que el viaje concluyó, también en formato numérico.
- **score:** Una puntuación asignada al viaje por la aplicación, reflejando diversos criterios de evaluación.
- **distance:** La distancia total recorrida durante el viaje, expresada en kilómetros y en formato numérico.
- **timeZone:** La zona horaria en la que se realizó el viaje, en formato de cadena de caracteres con un máximo de 128 caracteres.
- **user:** Identificación del usuario asociado al viaje, en formato de cadena de caracteres con un máximo de 48 caracteres.
- **numLocations:** Número de ubicaciones registradas durante el viaje, correspondiente al número de filas disponibles por viaje en otro tipo de archivo relacionado, en formato entero de 64 bits.
- **initialLocation** y **endLocation:** Ubicación inicial y final del viaje, respectivamente, ambas en formato de cadena de caracteres con un máximo de 128 caracteres, representadas por el nombre del municipio correspondiente.

- **initialZipCode** y **endZipCode**: Códigos postales de la ubicación inicial y final del viaje, en formato de cadena de caracteres con un máximo de 64 caracteres.
- **service:** Lista de servicios asociados al usuario, en formato de cadena de caracteres con un máximo de 256 caracteres, siendo este campo opcional.
- **visible:** Variable booleana que indica si el viaje es visible para otros usuarios o no, con el valor predeterminado de "true".
- **reason:** Motivo del viaje, en formato de cadena de caracteres con un máximo de 128 caracteres.
- **type:** Tipo de viaje, siendo "CAR" (automóvil) el valor predeterminado, en formato de cadena de caracteres con un máximo de 128 caracteres y es opcional.

2.1.2. *Dataset* de posiciones

En el *dataset* de posiciones, cada fila contiene información sobre una ubicación específica registrada durante los viajes. La información detallada que se registra en cada posición, según las columnas del archivo, incluye:

- **timestamp:** Presente en todas las filas. Indica el instante exacto en que se registraron los datos de la posición, en milisegundos desde el 1 de enero de 1970, en formato numérico.
- **latitude:** También presente en todas las filas. Muestra la latitud de la posición en grados decimales, en formato numérico de alta precisión.
- **longitude:** Aparece en todas las filas. Muestra la longitud en grados decimales, en formato numérico de alta precisión.
- **altitude:** Figura en el 99,95% de las filas. Representa la altitud en metros, en formato numérico.
- **horizontalAccuracy:** Aparece en el 99,99% de las filas. Indica la precisión horizontal del posicionamiento, en formato numérico de alta precisión.
- **verticalAccuracy:** Presente en el 99,19% de las filas. Muestra la precisión vertical, en formato numérico de alta precisión.
- **speed:** Figura en el 95,85% de las filas. Denota la velocidad en Km/h, en formato numérico.
- **speedAccuracy:** Aparece en el 98,37% de las filas. Indica la precisión de la velocidad medida, en formato numérico de alta precisión.

- **courseAccuracy:** Presente en el 92,68% de las filas. Denota la precisión del rumbo registrado, en formato numérico de alta precisión.
- **phoneUsage:** Figura en el 99,98% de las filas. Muestra si el teléfono se usó durante el registro de la posición; un valor mayor a 0 indica uso, dependiente del sistema operativo del móvil.
- **deviceOrientation:** Presente en el 97,50% de las filas. Indica la orientación del dispositivo durante el registro, con posibles valores como portrait, landscape, cara arriba, entre otros.
- **accelerationModule**, **accelerationX**, **accelerationY**, **accelerationZ:** Estas columnas, presentes en el 0,15% de las filas, indican la aceleración absoluta y en los ejes X, Y, Z respectivamente, en m/s^2, en formato numérico de alta precisión.
- **travel:** Aparece en todas las filas. Identifica el viaje al que pertenece la posición, formado por la concatenación de "initialTime" y "user" (presentes en el *dataset* de viajes), en formato de cadena con un máximo de 128 caracteres.
- **score:** Figura en el 0,23% de las filas. Representa los puntos que se restan por infracciones, en formato numérico con dos decimales.
- **date:** Presente en todas las filas. Fecha del viaje correspondiente, en formato date.
- **course:** Aparecen el 97,35% de las filas. Muestra el rumbo en grados, de 0 a 360 grados siendo 0 el norte, en formato numérico de alta precisión.
- **posMaxSpeed:** Presente en el 93,87% de las filas. Indica la velocidad máxima permitida en la vía, recuperada de un servicio externo, en formato numérico.
- **accelerationVelocity:** Figura en el 0,12% de las filas. Se registra cuando se excede un límite de aceleración o frenado, calculado por la app, en formato numérico.
- **differenceCourse:** Aparece en el 6,74% de las filas. Muestra la variación de rumbo en grados, en formato numérico con tres dígitos.

2.2. Descarga de datos, fuentes internas y externas

Los distintos *datasets* proporcionados para este trabajo no se presentan en el formato CSV convencional, sino que son archivos especiales de macOS utilizados para almacenar metadatos (Figura 3).

```
Datos-originales
|--Travels
|   --._bigquery_travels000000000000.csv
|   --._bigquery_travels000000000001.csv
|--Positions
|   --._bigquery_positions000000000000.csv
|   --._bigquery_positions000000000001.csv
|   --._bigquery_positions000000000002.csv
|   --._bigquery_positions000000000003.csv
|   --._bigquery_positions000000000004.csv
|   --._bigquery_positions000000000005.csv
|   --._bigquery_positions000000000006.csv
|   --._bigquery_positions000000000007.csv
|   --._bigquery_positions000000000008.csv
|   --._bigquery_positions000000000009.csv
```

Figura 3. Formato de los archivos proporcionados,

Estos archivos poseen estructuras únicas que incluyen información relevante como las URLs originales donde poder descargar los archivos CSV originales o datos de cuarentena, que son utilizados por macOS para administrar los archivos obtenidos de internet.

Dado que estos archivos no están en un formato directamente utilizable para un análisis, se desarrolló un *script* específico, recogido en el ANEXO A. CÓDIGO DESARROLLADO PARA DESCARGAR CSV. Este *script* se encarga de acceder a las URLs mencionadas, descargar la información relevante, y reubicar estos datos automáticamente desde el directorio de descargas hasta el directorio de trabajo (Figura 4).

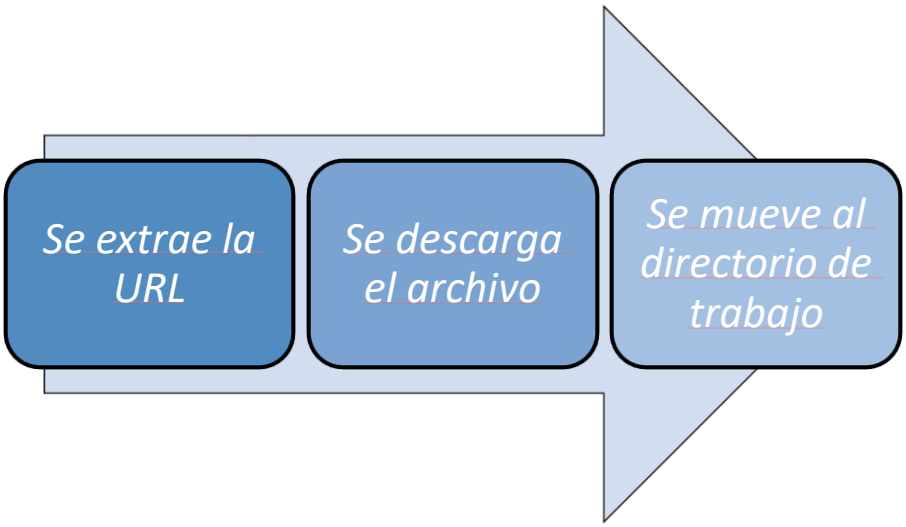


Figura 4. Proceso de descarga de los CSV.

Este proceso permite transformar los daos a un formato CSV estándar, facilitando así su posterior análisis y manejo.

2.3. Análisis de los datos

Una vez descargados los datos, se juntan todos los *datasets* de posiciones en un único *dataset* global de posiciones, y todos los *datasets* de viajes en un único *dataset* global de viajes.

Como se viene comentando, se cuenta con un total de 17351 viajes que fueron registrados desde marzo de 2022 hasta octubre de 2023 con el pico de viajes registrados en octubre de 2022 (Figura 5). La mayoría de los viajes fueron registrados a las 12, 16 y 17 horas, horario en el que la gente suele salir de trabajar (Figura 6).

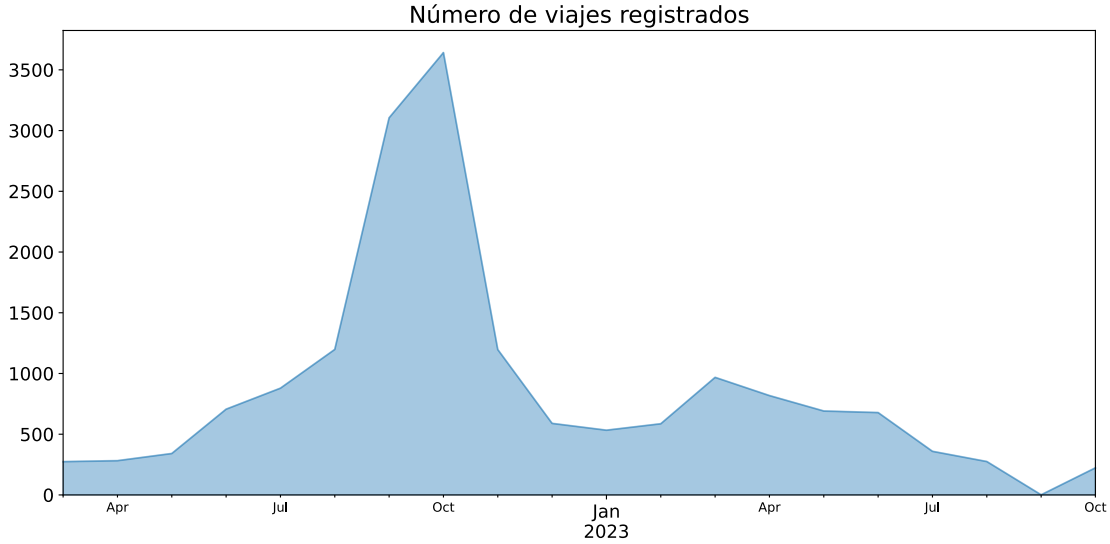


Figura 5. Número de viajes registrados en la aplicación.

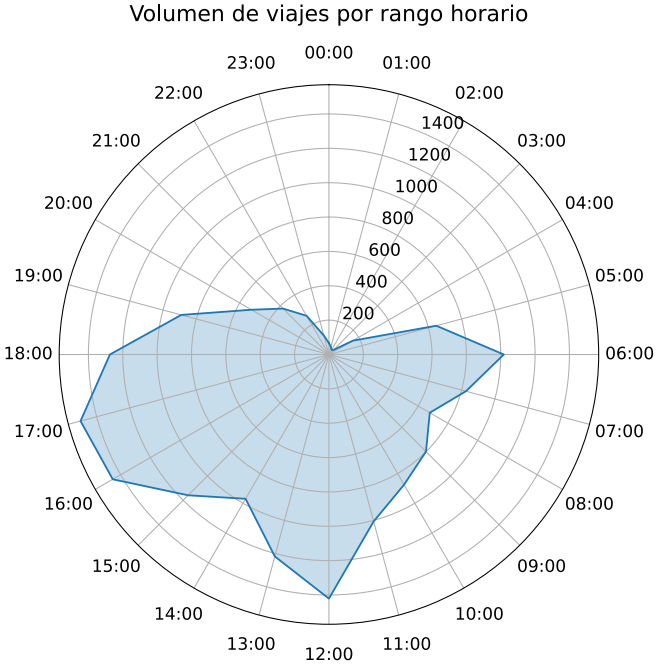


Figura 6. Volumen de viajes por rango horario.

Cabe destacar que, como era de esperar, el *dataset* de posiciones resulta ser significativamente más grande que el *dataset* de viajes (Figura 7).

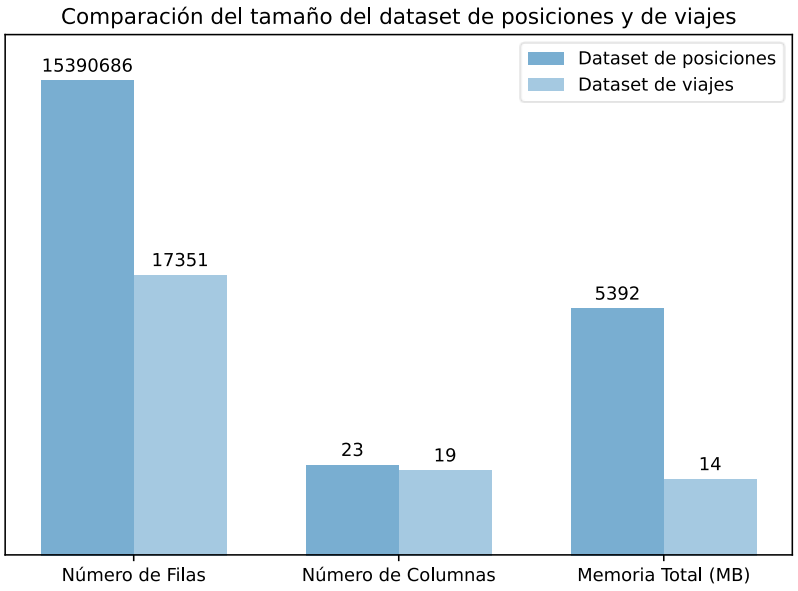


Figura 7. Comparación del tamaño del *dataset* de posiciones y de viajes en escala logarítmica.

Para obtener una comprensión más profunda de la naturaleza de ambos *datasets*, se realiza un análisis preliminar básico. En este análisis se observan las principales propiedades estadísticas de los datos, así como el número de variables categóricas presentes (Figura 8) o la cantidad de registros nulos, duplicados y válidos (Figura 9).

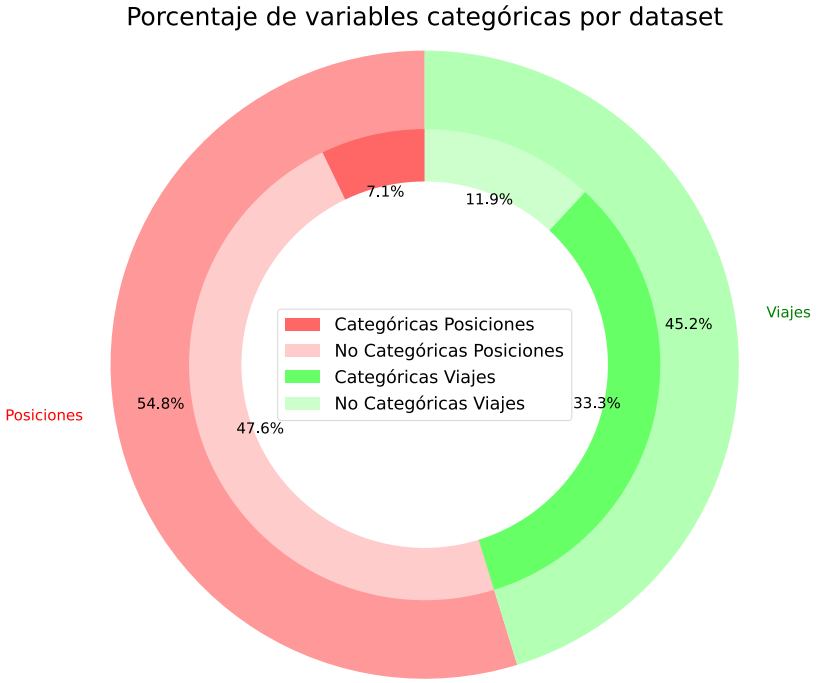


Figura 8. Porcentaje de variables categóricas en el *dataset* de posiciones y de viajes.

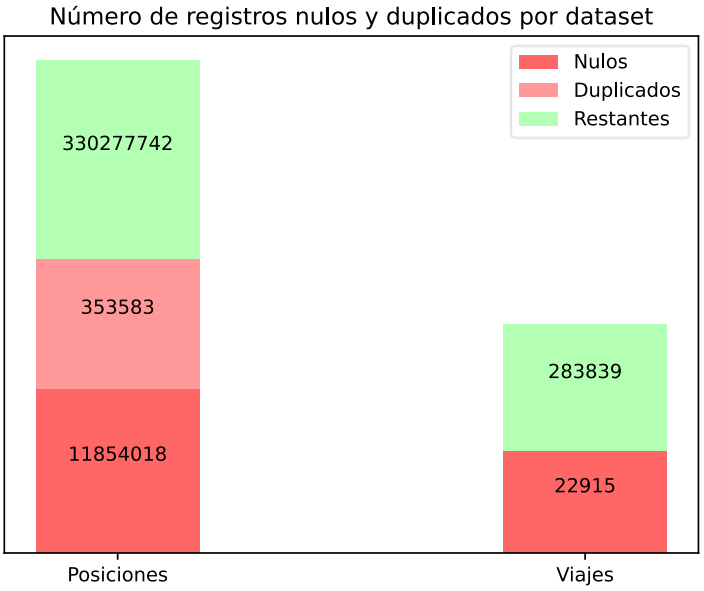


Figura 9. Número de registros nulos y duplicados en el *dataset* de posiciones y de viajes.

2.4. Limpieza y normalización de datos

La calidad de los datos es fundamental para asegurar la validez de cualquier análisis en la investigación científica. En el ámbito de la caracterización de estilos de conducción, donde la precisión y la representatividad de los datos son cruciales, la limpieza y normalización de los mismos se convierten en una etapa esencial del proceso de análisis. Este capítulo describe detalladamente los métodos y procedimientos adoptados para preparar el conjunto de datos para su posterior análisis, asegurando que la información sea fiable y adecuada para identificar patrones de conducción.

El proceso de limpieza de datos se estructura en tres fases sucesivas:

Fase I: Esta fase se centra en la eliminación de las columnas que no aportan información relevante. Asimismo, se eliminan anomalías simples, tales como valores faltantes o claramente erróneos, además de eliminar todo aquel viaje que no haya sido registrado en coche.

Fase II: En esta sección se realiza una inspección más rigurosa para identificar y corregir problemas menos evidentes, como inconsistencias en los datos que, aunque no presentan errores obvios, podrían afectar la calidad de apartados posteriores.

Fase III: En esta etapa se eliminan aquellos viajes que no están duplicados como tal, pero que son considerados redundantes porque representan parte de otro viaje registrado momentos después por el copiloto.

Cada una de estas fases contribuye de manera significativa a la robustez del conjunto de datos, preparándolos para un análisis confiable y detallado de los estilos de conducción.

2.4.1. Fase I

En primer lugar, se concatenan el *dataset* de posiciones y de viajes. Para ello, se crea una columna adicional "travel" en el *dataset* de viajes sobre la que se hará el *join* de ambos *DataFrames*. Dicha columna es un identificador único de cada viaje generado por la combinación de la columna "initialTime", un símbolo de suma "+" y el identificador único del usuario, columna "user".

El *dataset* contiene viajes registrados en coche, tren y avión. Dado que el objetivo de este trabajo es clasificar tipos de conducción, se ha decidido prescindir de los viajes que no han sido registrados en coche (Figura 10).

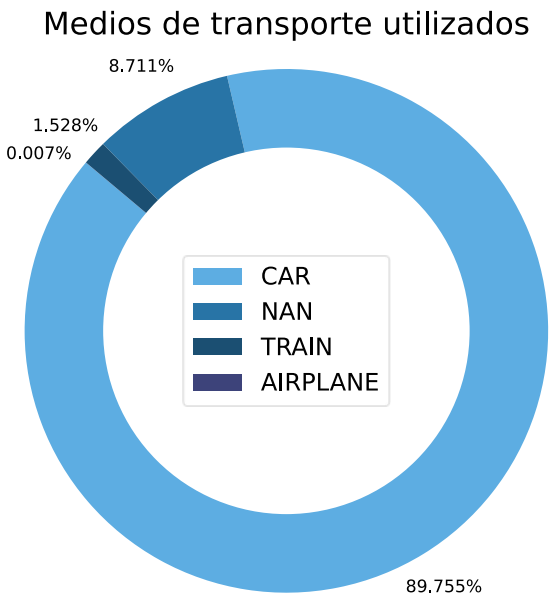


Figura 10. Medios de transporte utilizados en los distintos viajes.

Tras quedarnos sólo con los viajes registrados en coche, y una vez que se ha unificado el conjunto de datos global, se procede a eliminar tanto los registros duplicados como los que contengan valores nulos. Aunque es viable imputar una cantidad significativa de registros, la amplia cantidad de datos disponibles hace que esta medida no se considere necesaria.

Por otro lado, con el objetivo de representar los distintos viajes de una forma visual e intuitiva, se han generado distintas figuras en las que se compara la velocidad con la que circula el usuario a lo largo del viaje frente a la velocidad máxima permitida en la vía (Figura 11).

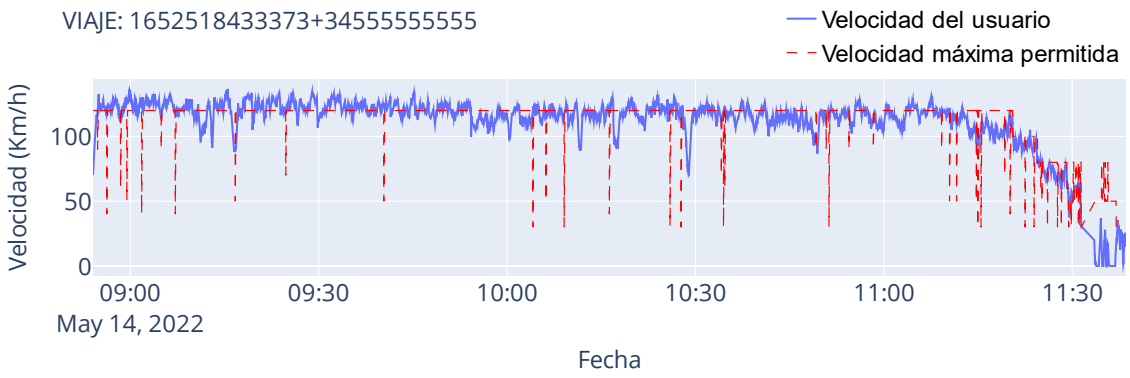


Figura 11. Representación de la velocidad del usuario y de la velocidad máxima permitida en la vía para un determinado viaje.

Tal y como se puede observar, no es una representación del todo fiable. Hay una cantidad nada despreciable de picos en los que se produce abruptamente un decremento de la velocidad máxima permitida en la vía. A continuación, se muestra un fragmento concreto del *DataFrame* donde se puede apreciar este efecto (Figura 12).

	date	latitude	longitude	speed	phoneUsage	travel_id	course	maxSpeed
2639	2022-05-14 09:40:20	40.500544	0.416006	124.160000	0.000000	1652518433373+34555555555	2.000000	120.000000
2640	2022-05-14 09:40:21	40.500856	0.416020	124.340000	0.000000	1652518433373+34555555555	1.000000	120.000000
2641	2022-05-14 09:40:22	40.501167	0.416026	124.090000	0.000000	1652518433373+34555555555	0.000000	50.000000
2642	2022-05-14 09:40:24	40.501787	0.416027	123.840000	1.000000	1652518433373+34555555555	0.000000	120.000000
2643	2022-05-14 09:40:25	40.502096	0.416025	123.980000	0.000000	1652518433373+34555555555	359.000000	120.000000

Figura 12. Fragmento del *DataFrame* donde se puede apreciar un pico abrupto de decremento de velocidad máxima permitida.

Esto sucede porque en un viaje hay muchas intersecciones donde se detecta erróneamente el valor de la velocidad máxima permitida en la vía. Por ejemplo, si se buscan en Google Maps las coordenadas de la fila 2641 del *DataFrame* se puede observar la Figura 13.

Figura 13. Intersección entre una autovía y una carretera secundaria (Fuente: Google Maps).

En este caso, se está detectando la velocidad máxima permitida en la carretera secundaria en vez de la autovía. Para solucionar este problema, se ha desarrollado una función que tiene como objetivo suavizar la serie temporal de velocidades máximas, especialmente corrigiendo cambios bruscos como los descritos anteriormente.

Dicha función tiene como objetivo recorrer la columna maxSpeed de un viaje en concreto y comparar el valor actual con el anterior para detectar si ha habido un cambio brusco en la velocidad máxima permitida en la vía. En caso de encontrarlo, recorre las siguientes tres posiciones para asegurar que no ha sido un cambio puntual, sino que se mantiene a lo largo del tiempo (Figura 14).

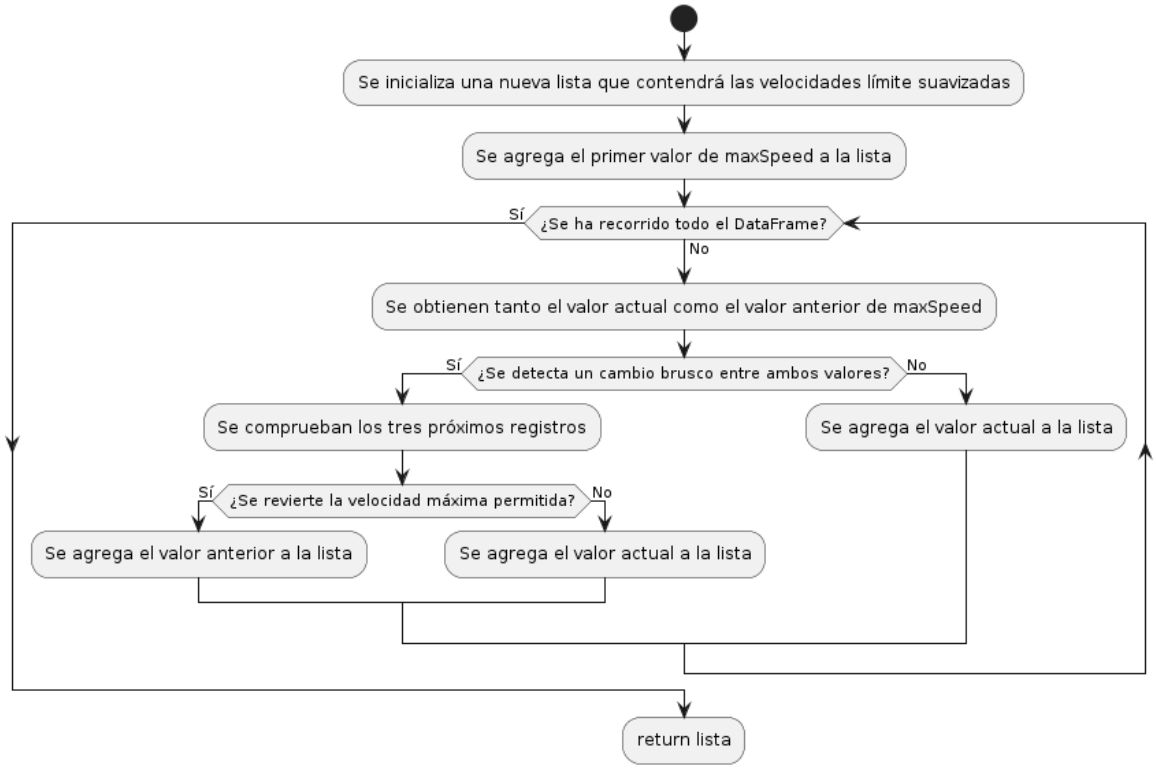


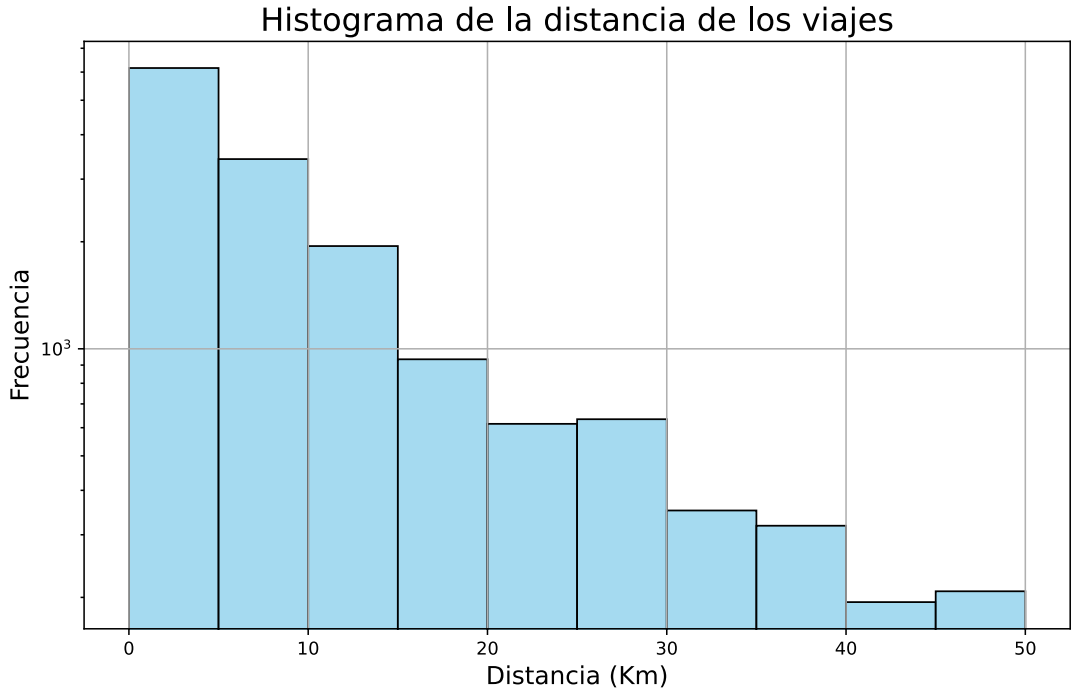
Figura 14. Función utilizada para suavizar los picos erróneos de velocidad máxima.

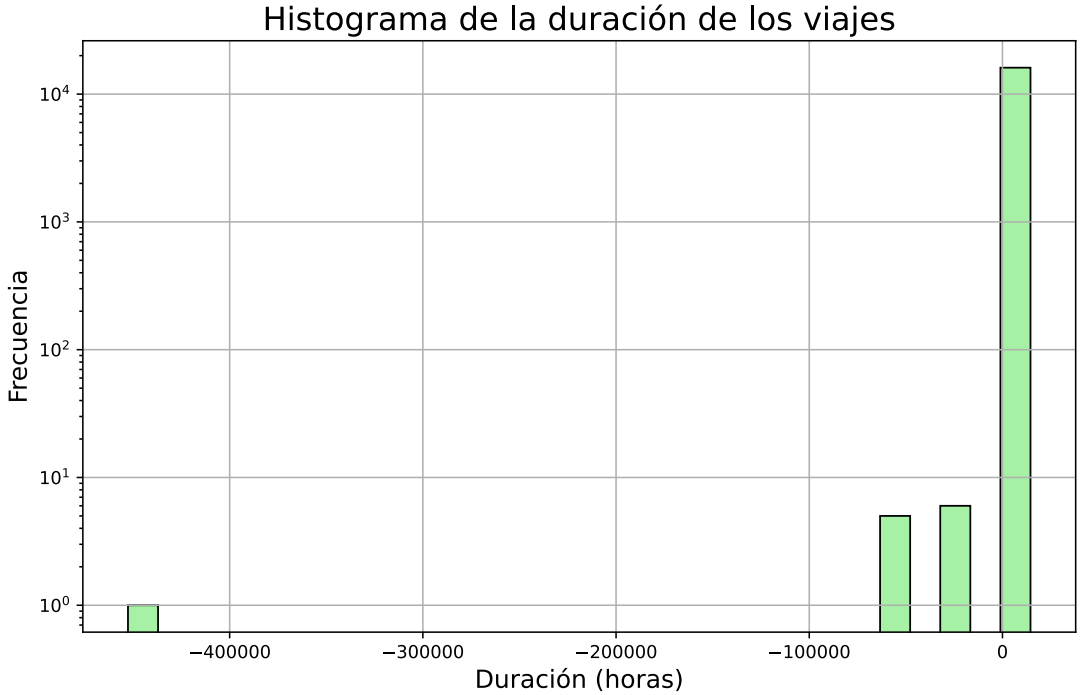
Tras aplicar dicha función al *DataFrame* se obtienen viajes notablemente más limpios (Figura 15).

Figura 15. Resultado de aplicar la función de suavizado a la Figura 11.

2.4.2. Fase II

En esta fase se busca detectar valores atípicos más difíciles de percibir a simple vista. Para ello, en primer lugar, se representan histogramas de la distancia, duración y número de puntos registrados de los viajes (Figura 16).





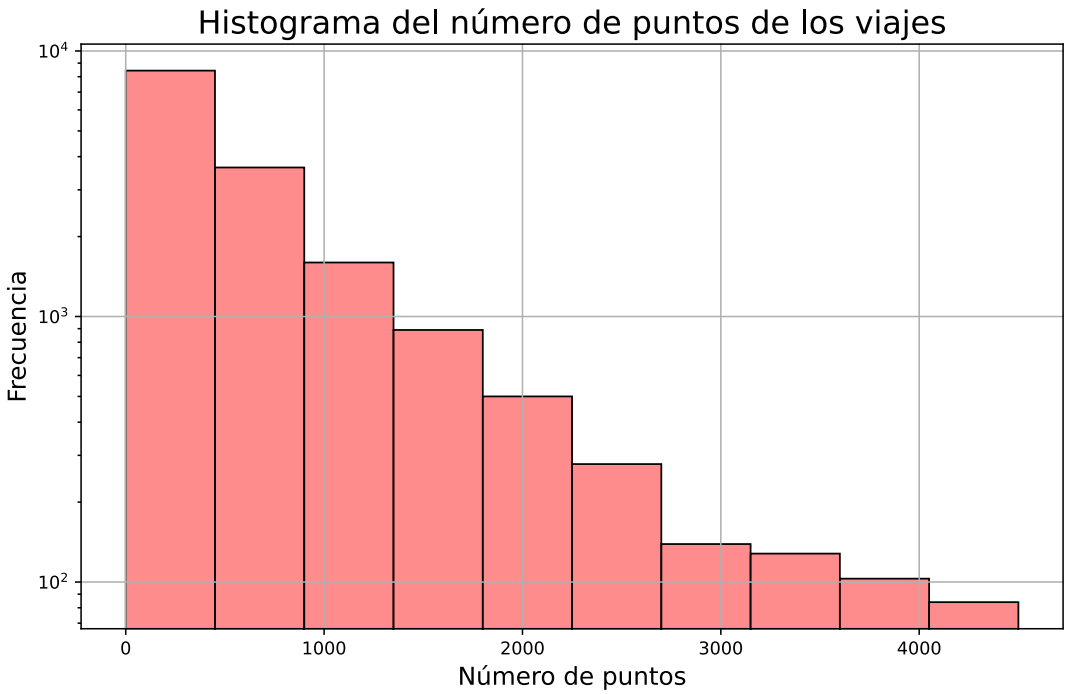


Figura 16. Representación en escala logarítmica de histogramas de la distancia, duración y número de puntos registrados de los viajes.

Se puede observar que hay una serie de viajes que, por fallos de la aplicación de la empresa externa, tienen duración negativa. Todos esos viajes erróneos se limpian del *dataset*. Por otro lado, cabe destacar que la mayoría de los viajes son de duración relativamente corta y cuentan con pocos puntos. Para facilitar la visualización y comparación de los datos, los histogramas se han representado en escala logarítmica. Esto permite una mejor interpretación visual cuando se comparan valores muy pequeños con otros significativamente más altos, asegurando que ambos extremos se muestren de manera clara y comprensible.

El problema de los viajes cortos o con pocos puntos es que aportan muy poca información (Figura 17).

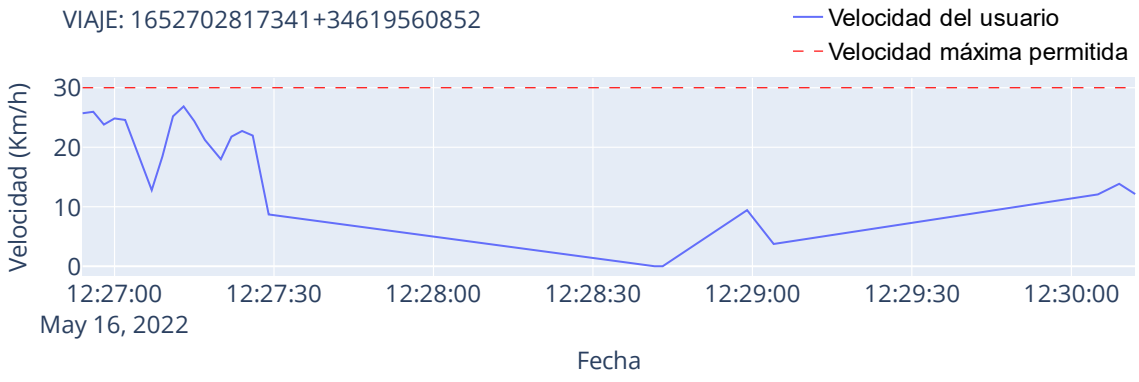


Figura 17. Representación de un viaje corto.

Por este motivo, se eliminan todos los viajes con una distancia menor o igual que 10 Km y con un número de puntos menor que 900.

Por otro lado, se han encontrado viajes con velocidades negativas durante un tiempo prolongado (Figura 18).

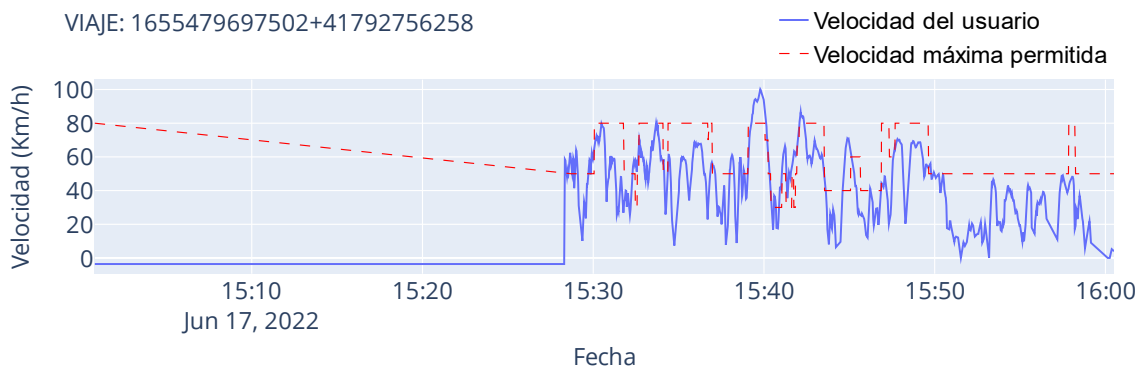


Figura 18. Representación de un viaje con velocidad negativa.

Se puede llegar a considerar que una velocidad negativa puede representar marcha atrás en un determinado instante, pero si es durante un tiempo relativamente largo es producto de un fallo de la aplicación, por lo que se decide eliminar esos valores (Figura 19).

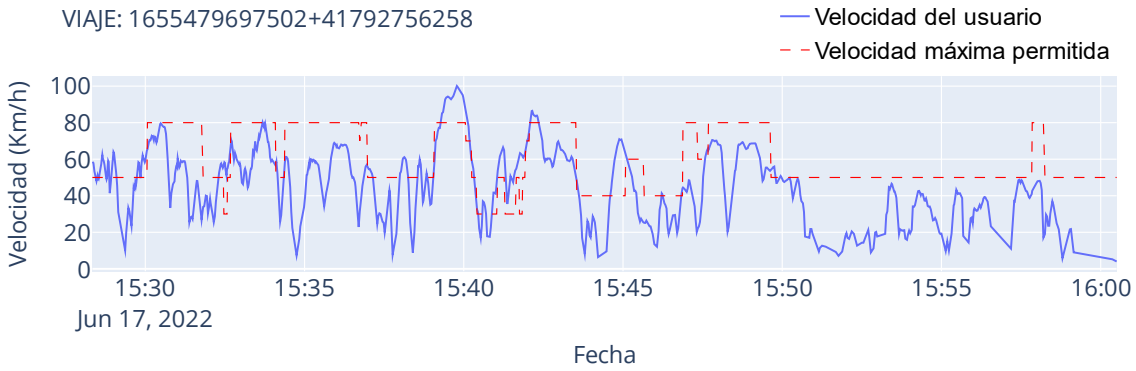


Figura 19. Representación de la Figura 18 tras haber eliminado los valores negativos de velocidad.

2.4.3. Fase III

En esta fase, principalmente, se eliminan viajes que han sido registrados por más de un usuario a la vez, por lo que aportan información duplicada y redundante (Figura 20).

	date	latitude	longitude	speed	phoneUsage	travel_id	course	maxSpeed
29080	2022-05-16 18:45:04.000	39.533010	-1.908680	111.82	0.0	1652721076445+34555555555	248.00	120.0
29081	2022-05-16 18:45:05.000	39.532911	-1.909019	111.05	0.0	1652721076445+34555555555	248.97	120.0
29082	2022-05-16 18:45:06.000	39.532806	-1.909355	112.35	0.0	1652721076445+34555555555	246.04	120.0
29083	2022-05-16 18:45:07.000	39.532688	-1.909684	112.39	0.0	1652721076445+34555555555	245.00	120.0
29084	2022-05-16 18:45:08.000	39.532568	-1.910014	111.92	0.0	1652721076445+34555555555	244.00	120.0

	date	latitude	longitude	speed	phoneUsage	travel_id	course	maxSpeed
37038	2022-05-16 18:45:04	39.533010	-1.908659	112.65	0.0	1652723087803+34444444444	248.00	120.0
37039	2022-05-16 18:45:05	39.532905	-1.908997	114.75	0.0	1652723087803+34444444444	247.02	120.0
37040	2022-05-16 18:45:06	39.532795	-1.909340	114.29	0.0	1652723087803+34444444444	247.00	120.0
37041	2022-05-16 18:45:07	39.532681	-1.909673	112.29	0.0	1652723087803+34444444444	245.05	120.0
37042	2022-05-16 18:45:08	39.532563	-1.909997	111.86	0.0	1652723087803+34444444444	245.00	120.0

Figura 20. *DataFrames* de viajes registrados por más de un usuario.

A continuación, en la Figura 21, se muestran ambas series temporales. Gráficamente se puede apreciar incluso más fácilmente que el segundo viaje es parte del primero.

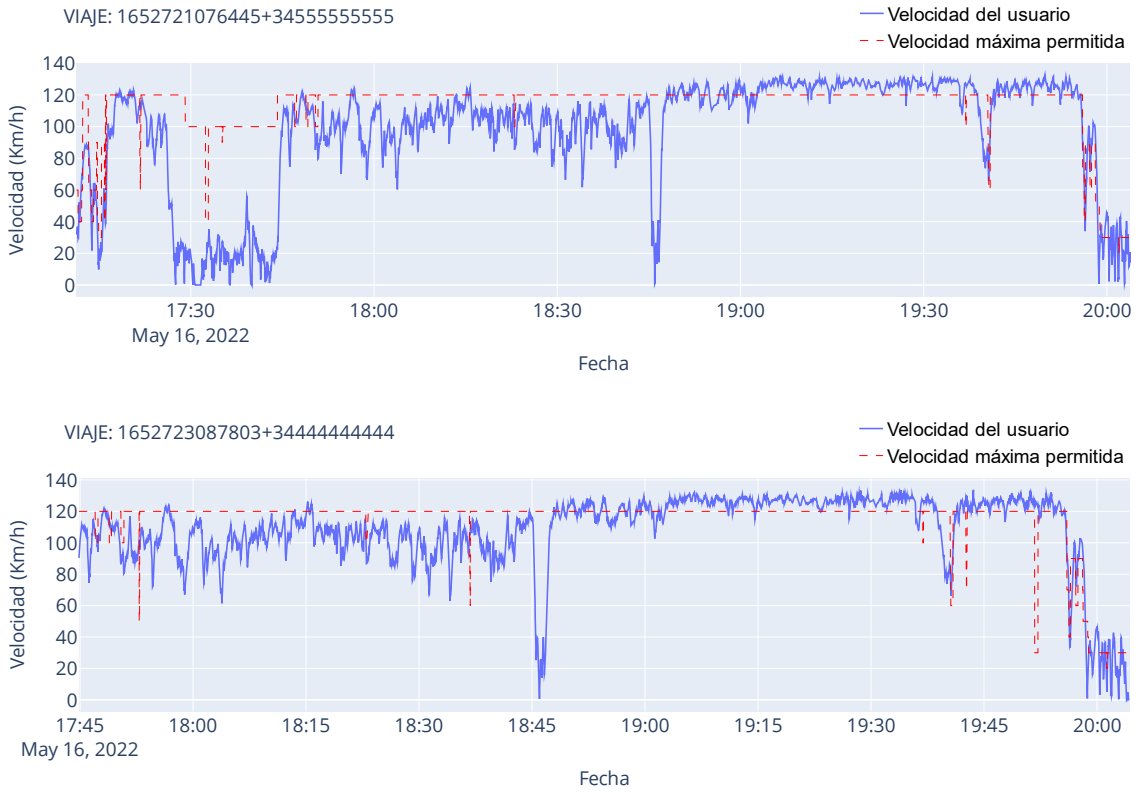


Figura 21. Representación gráfica de los *DataFrames* redundantes.

Para limpiar estos viajes, se ha desarrollado una función que, en caso de encontrar viajes redundantes, se queda con el que mayor número de puntos tiene, puesto que es el que más información puede llegar a aportar (Figura 22).

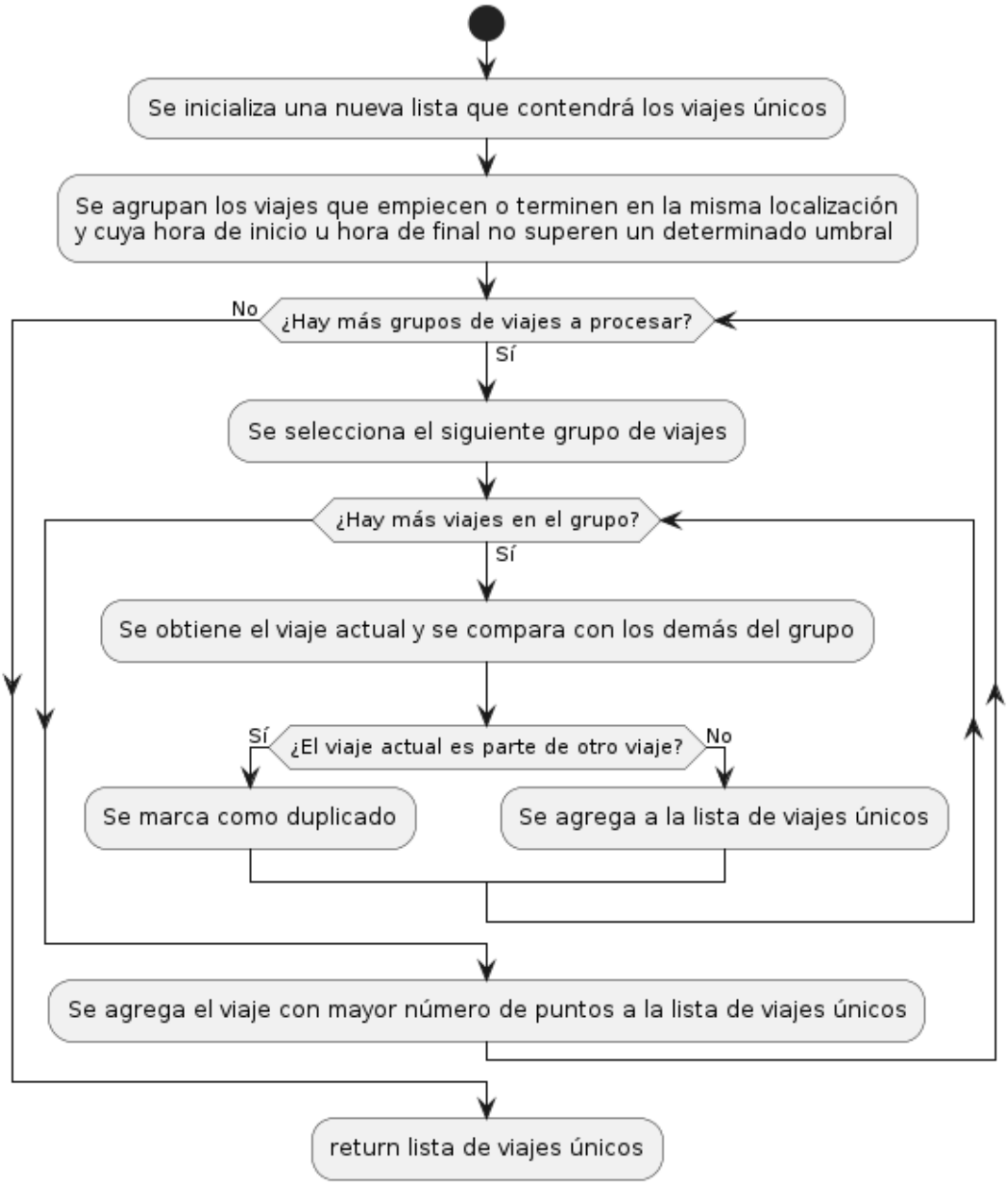


Figura 22. Función utilizada para elimiinar viajes redundantes.

2.4.4. Resumen del proceso de limpieza

En este apartado se representa gráficamente la transformación de los datos a lo largo de todo el proceso de limpieza. Se han reducido las filas un 50% (Figura 23) y el uso de memoria un 87% (Figura 24).

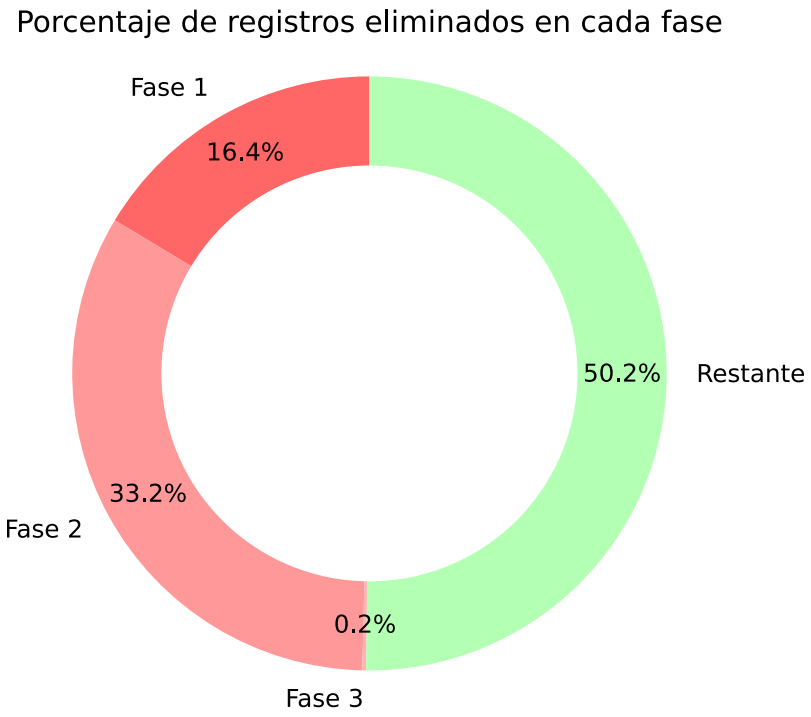


Figura 23. Registros eliminados en cada fase.

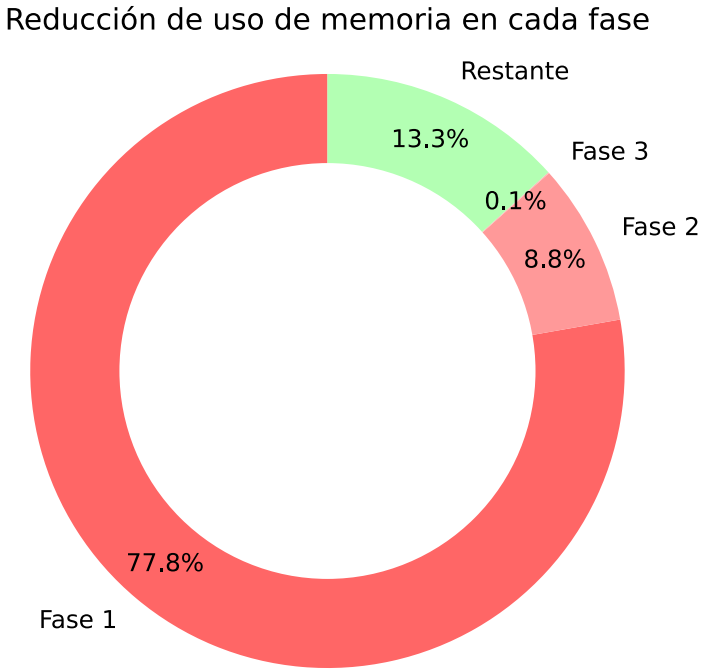


Figura 24. Reducción de uso de memoria.

Finalmente, en la Figura 25 se compara la reducción del número de filas y columnas en cada fase y cómo repercute este hecho en el uso de memoria. Se puede observar que la limpieza realizada en la fase 3 es prácticamente imperceptible.

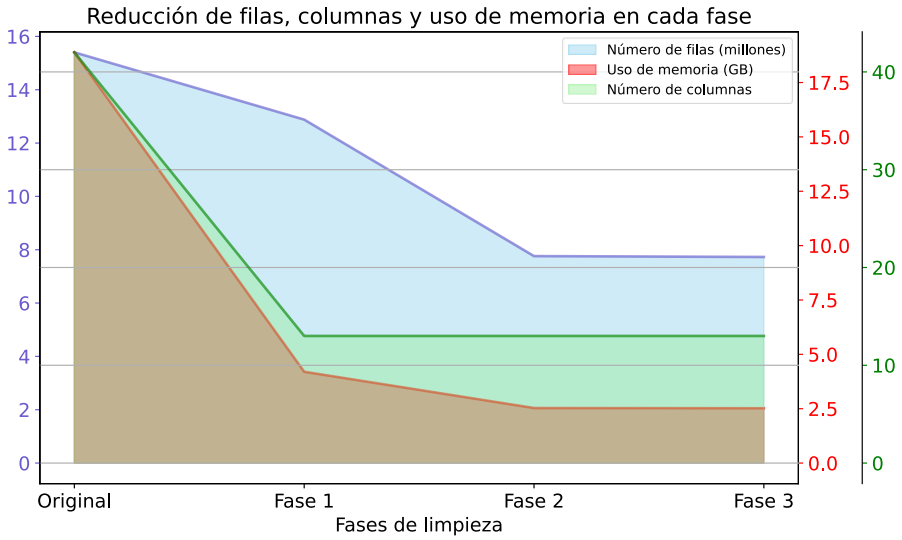


Figura 25. Reducción de filas, columnas y uso de memoria.

2.5. Feature engineering

Una vez completado el proceso de limpieza de datos, estamos en condiciones de proceder con el *clustering*. Debido a la existencia de múltiples entradas por cada viaje, es esencial resumir toda esa información en una sola entrada para poder realizar una agrupación efectiva mediante *clustering*. Para este propósito, se han propuesto tres características (*features*):

1) **time_over_maxSpeed:** Representa el porcentaje de tiempo durante el cual el usuario excede la velocidad máxima permitida. Esta métrica permite evaluar la tendencia del conductor a superar los límites de velocidad establecidos.
2) **phone_usage:** Mide el porcentaje de tiempo en que el conductor utiliza el teléfono móvil mientras conduce. Este dato es crucial para analizar comportamientos de distracción al volante.
3) **hard_acceleration:** Indica el porcentaje de tiempo en que el usuario realiza aceleraciones bruscas, lo que permite identificar patrones de conducción agresiva.

Obtener esta última característica es un procedimiento más laborioso que el realizado con las otras dos características. En este caso, se calculan tanto la aceleración longitudinal como la aceleración lateral (Ecuaciones (1) y (2)) [28], [29].

$$acc_{LONG} = \frac{(v_n - v_{n-1})}{(t - t_{n-1})} \quad (1)$$

$$acc_{LAT} = \frac{v^2}{R} = v\omega \tag{2}$$

Donde la velocidad angular se calcula con la variación de la orientación (Ecuación (3)) [30] (columna course del *DataFrame*).

$$\omega = \frac{\partial \theta}{\partial t} \tag{3}$$

En un estudio previo desarrollado por Centro Zaragoza, se ha elaborado una tabla (Tabla 1) en la que se definen una serie de umbrales (siendo g la fuerza de la gravedad) para distintas aceleraciones, con el fin de caracterizar determinados estilos de conducción [31].

Tabla 1. Umbrales de aceleración en función del tipo de conducción.

ACELERACIÓN	ECO	NORMAL	AGRESIVO
LATERAL	$\|a\| < 0.4g$	$0.4g < \|a\| < 0.7g$	$\|a\| > 0.7g$
LONGITUDINAL (desde parada)	$\|a\| < 0.3g$	$0.3g < \|a\| < 0.5g$	$\|a\| > 0.5g$
LONGITUDINAL (en circulación)	$\|a\| < 0.12g$	$0.12g < \|a\| < 0.24g$	$\|a\| > 0.24g$

A continuación, en la Figura 26, se representan en escala logarítmica las aceleraciones longitudinales (arriba) y laterales (abajo) calculadas en circulación en un determinado viaje.

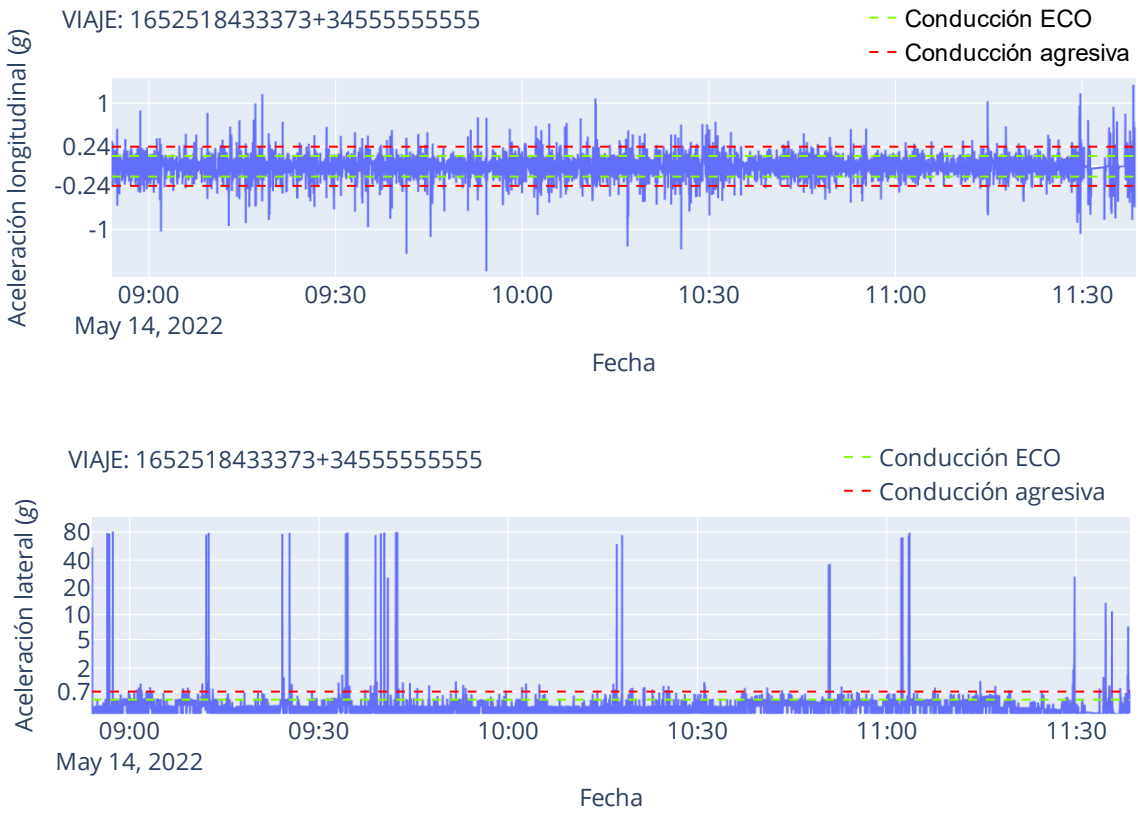


Figura 26. Aceleraciones longitudinales y laterales de un determinado viaje representadas en escala logarítmica.

En la gráfica de la aceleración lateral se observan una serie de picos espurios que carecen de sentido físico. Esto sucede porque se detecta una variación enorme entre 360 grados y 1 grado, cuando la diferencia realmente es mínima. Tras corregir este error se obtiene la Figura 27.

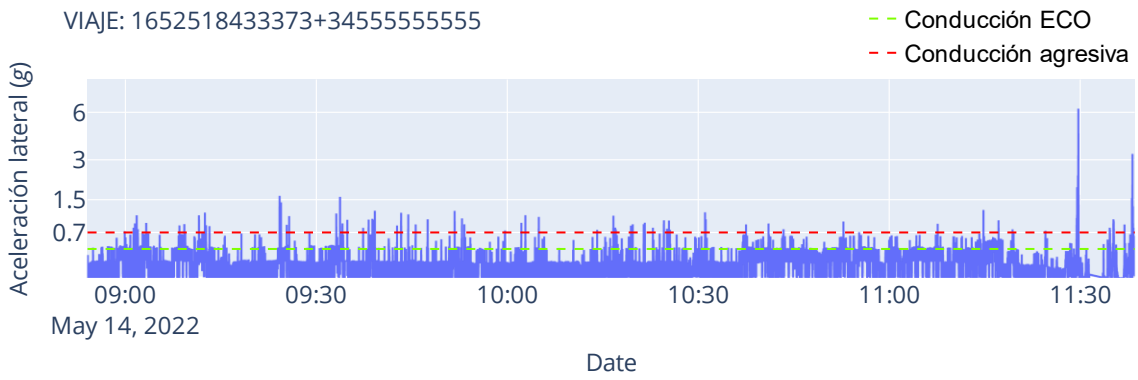


Figura 27. Aceleración lateral corregida representada en escala logarítmica.

Se puede observar que todavía siguen apareciendo picos excesivamente altos. Esto se puede deber a que, generalmente, el usuario no dispone de un soporte que asegure la sujeción del dispositivo móvil en todo momento, lo que podría provocar su caída del asiento o del lugar donde esté colocado. Dado que la cantidad de picos espurios es pequeña y que, en caso de ocurrir esta caída, implica una aceleración relativamente brusca, no se ha considerado necesario filtrarlos.

Cabe destacar que, aunque se representen en la gráfica tanto el umbral ECO como el agresivo, por simplicidad y coherencia con el resto de *features* que se han extraído, se ha considerado únicamente el umbral de estilo agresivo en circulación para obtener la *feature* "hard_acceleration".

Por otro lado, hay usuarios que circulan por encima del límite de velocidad durante todo el viaje. Lógicamente no es lo mismo pasarse por 5 Km/h yendo a 30 Km/h, que ir a 200 Km/h como se muestra en el viaje de la Figura 28. Este hecho puede afectar también al uso del móvil, puesto que, aunque en ambos casos el uso del móvil es un acto imprudente, el usuario que circula a 200 Km/h lo usará menos porque tiene un menor control sobre el coche y el entorno.

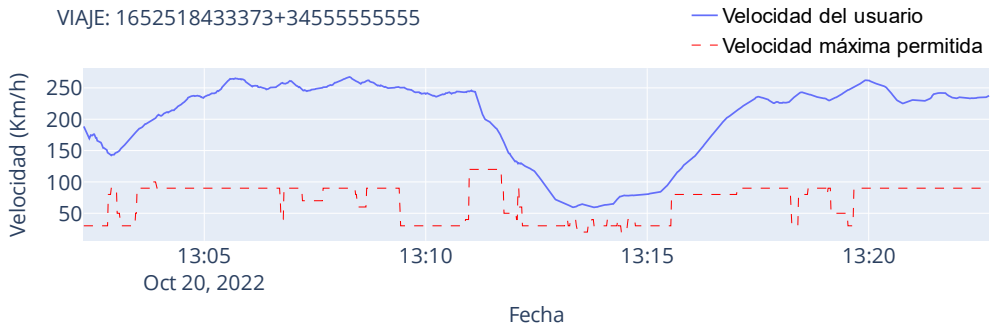


Figura 28. Representación de un viaje en el que se supera todo el tiempo la velocidad máxima permitida en la vía.

Por último, como paso inmediatamente anterior a la realización del clustering, y dado que se tienen distintos viajes realizados por el mismo usuario, se calcula la media y desviación típica de las distintas *features*.

3. RESULTADOS

En este apartado, se presentan los resultados obtenidos tras la aplicación de una serie de algoritmos de clustering. Se describen los algoritmos que finalmente han sido utilizados para modelar los perfiles de conducción definitivos, se detalla el procedimiento llevado a cabo para elegir el número óptimo de clusters y, por último, se muestra gráficamente la agrupación de clusters resultante.

3.1. Descripción de los algoritmos utilizados

Aunque se ha trabajado con múltiples algoritmos, en este apartado se hace una descripción de los algoritmos finalmente utilizados para modelar el perfil definitivo de los conductores; k-Means y Clustering Espectral.

3.1.1. k-Means

3.1.1.1 Descripción

k-Means es la técnica de clustering más popular. Se trata de un algoritmo basado en particiones, es decir, trata de descubrir agrupaciones de los datos optimizando una función objetivo de tal manera que en cada iteración se mejora la calidad de las particiones [32].

El algoritmo de k-Means divide un conjunto inicial de n elementos en k clusters distintos que quedan descritos por su valor medio, conocido como centroide.

Cada cluster representa un grupo de elementos afines y a la medida dispersión de cada cluster es lo que se denomina inercia o coherencia del cluster. Esta medida se calcula como la suma del cuadrado de las distancias de cada elemento x_i al centroide c_j correspondiente normalizada por el número de elementos del cluster n_k (Ecuación (4)):

$$inertia = \frac{1}{n_k} \sum_{x_i \in C_k} \left\| x_i - c_j \right\|^2 \tag{4}$$

Cada centroide se calcula como el valor medio: $c_k = \frac{\sum x_i}{n_k}$

Para todos los clusters de un *dataset* de *n* elementos, la inercia total se representa por la medida del error que quiere minimizarse (*Sum of Squares Error*, SSE (Ecuación (5)).

$$SSE(C) = \frac{1}{n} \sum_{j=1}^{k} \sum_{x_i \in C_k} \left\| x_i - c_j \right\|^2 \tag{5}$$

3.1.1.2 Funcionamiento

Una vez definidos estos conceptos previos, se procede a explicar brevemente el funcionamiento del algoritmo:

- Se seleccionan *k* centroides al azar que no tienen por qué corresponder con puntos del *dataset*.
- Se forman *k* clusters asignando a cada punto el centroide más próximo (generalmente se utiliza la distancia Euclideana).
- Se recalculan los centroides de cada cluster hasta que los centroides no cambien o que se cumpla con un criterio de convergencia.

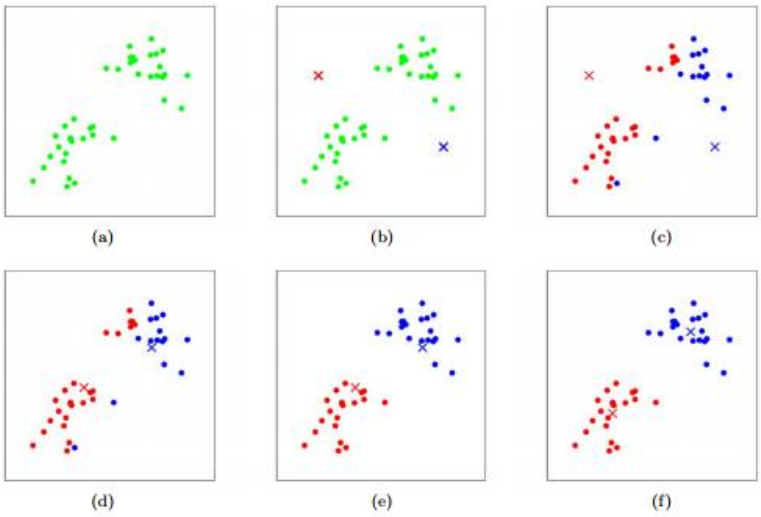


Figura 29. Resumen gráfico del funcionamiento de k-Means. (a) *Dataset* original. (b) Centroides aleatorios iniciales. (c-f) Representación de ejecutar dos iteraciones de k-Means. (Fuente: [32]).

3.1.1.3 Limitaciones

Se asumen formas hiperesféricas de los *clusters* y de tamaños semejantes. En caso de tener *clusters* con formas no convexas existen otros algoritmos más adecuados, por ejemplo, DBSCAN.

El número de *clusters* se debe seleccionar a priori. Existen otros métodos, como *Affinity Propagation*, donde no se necesita calcular o estimar el número de *clusters*.

Es sensible a la presencia de *outliers*. Para solucionar este problema se podría optar por utilizar algoritmos más robustos como k-Medoids.

3.1.1. Clustering Espectral

3.1.1.1 Descripción

El *clustering* espectral es una técnica de agrupamiento basada en la teoría espectral de grafos. Esta técnica utiliza el espectro (valores propios) de la matriz de similitud del *dataset* para realizar la reducción dimensional antes de agrupar en menos dimensiones. Es particularmente útil para identificar estructuras en datos no lineales [33].

El algoritmo de clustering espectral se basa en la representación de los datos como un grafo, donde los nodos representan los puntos de datos y las aristas representan la similitud entre ellos. A través del análisis de este grafo, se pueden identificar grupos de nodos que están altamente conectados entre sí y menos conectados con los nodos de otros grupos.

3.1.1.2 Funcionamiento

Para entender el funcionamiento del clustering espectral, es esencial desglosar el proceso en una serie de pasos clave que transforman los datos originales en un espacio de alta dimensión a una representación más manejable y estructurada. A continuación, se describen los pasos principales involucrados en este método:

- **Construcción de la matriz de similitud:** Se construye una matriz S donde cada elemento Sij representa la similitud entre los puntos xi y xj. Esta similitud puede ser calculada utilizando distintas métricas, como la distancia euclidiana o el kernel gaussiano.
- **Cálculo de la matriz Laplaciana:** A partir de la matriz de similitud se construye la matriz Laplaciana L. Existen varias definiciones para la matriz Laplaciana, siendo la más común L=D-S, donde D es la matriz diagonal de grados.
- **Descomposición espectral:** Se realiza la descomposición en valores propios de la matriz Laplaciana para obtener los k primeros vectores propios. Estos vectores forman una nueva representación de los datos en un espacio de menor dimensión.
- **Clustering:** Finalmente, se aplica un algoritmo de *clustering* (como k-means sobre los vectores propios obtenidos para identificar los *clusters* en el nuevo espacio reducido.)

3.1.1.3 Limitaciones

La construcción de la matriz de similitud es crucial y puede afectar significativamente los resultados del *clustering*.

La descomposición espectral puede ser costosa para *datasets* grandes, ya que implica cálculos sobre matrices de gran tamaño.

Al igual que en k-means, se debe especificar el número de *clusters* a priori.

3.2. Discusión

A priori, Centro Zaragoza plantea la idea de que los conductores pueden clasificarse en dos grupos; prudentes e imprudentes. Para tratar de confirmar que dicha estructura es razonable y permite clasificar correctamente a los conductores, inicialmente se trabaja con el algoritmo de *clustering* más popular;

k-Means. Cabe destacar que no tiene por qué ser el número óptimo de *clusters*, se trata simplemente de una pequeña exploración preliminar.

A continuación, en la Figura 30, se puede identificar claramente un grupo más imprudente que el otro en todas las *features* extraídas.

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.30229 ± 0.11683	0.07431 ± 0.0997	0.21349 ± 0.04746
1	0.50162 ± 0.1208	0.1198 ± 0.13948	0.3014 ± 0.04478

Figura 30. Media y desviación típica de las tres *features* en cada uno de los *clusters* (k-Means).

Por otro lado, en la Figura 31 se puede apreciar una agrupación relativamente buena a pesar de la dispersión de los datos.

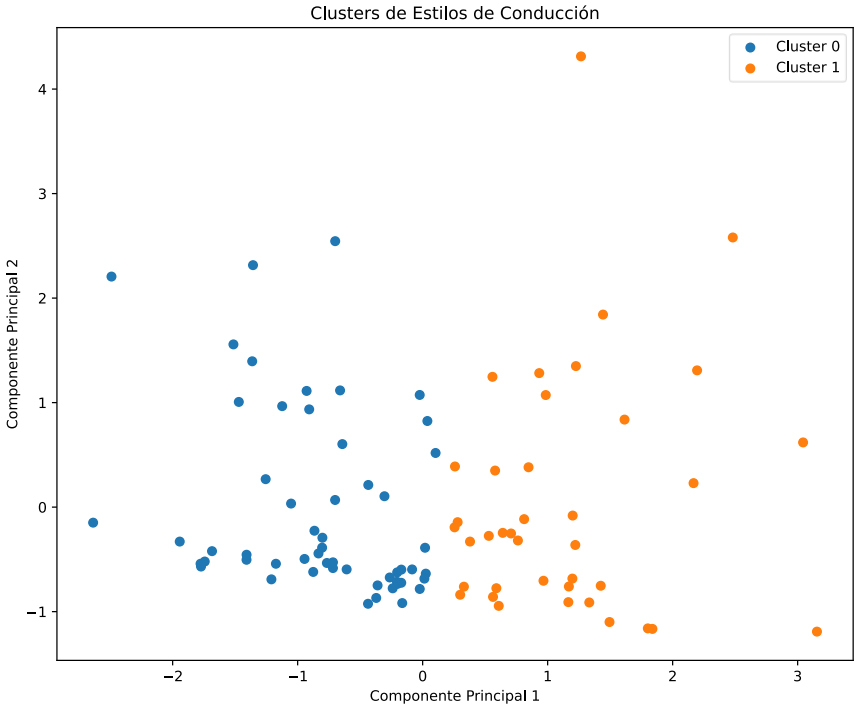


Figura 31. Representación gráfica de los dos *clusters* (k-Means).

Cabe destacar que en el análisis presentado, se emplea el método de Análisis de Componentes Principales (PCA) como una técnica complementaria para la exploración de los datos y la visualización de los resultados de la agrupación realizada con el algoritmo de k-Means. El PCA permite reducir la dimensionalidad de los datos, transformando las características originales en nuevas variables (componentes principales) que capturan la mayor parte de la varianza de los datos.

En este caso, el PCA facilita la representación de los clústeres en un espacio bidimensional, como se muestra en la Figura 31. Las dos primeras componentes principales se utilizan como ejes en la gráfica, permitiendo identificar visualmente la separación entre los dos clústeres (prudentes e imprudentes). Esta visualización es particularmente útil para validar de forma preliminar la calidad de la agrupación y para analizar patrones en los estilos de conducción. A pesar de haber cierta dispersión, la separación entre los grupos es claramente distinguible, lo que sugiere que las características seleccionadas para el análisis son relevantes y discriminativas.

Una vez realizado este primer análisis exploratorio, se sigue una metodología más robusta para identificar el número óptimo de *clusters* con la ayuda del método del codo, el índice de Calinski-Harabasz y el coeficiente de Silhouette.

3.2.1. Método del codo

Este método consiste en realizar el *clustering* para diferentes valores de *k* (número de *clusters*) y calcular la inercia para cada *k*. La idea es encontrar el punto donde la disminución de la inercia empieza a hacerse menos pronunciada. Este punto, que recuerda a un codo en un gráfico, indica el número óptimo de *clusters*, ya que añadir más *clusters* después de este punto no mejora significativamente la variabilidad explicada por el modelo.

A continuación, en la Figura 32, se muestra el resultado tras aplicar este método al problema expuesto en este trabajo.

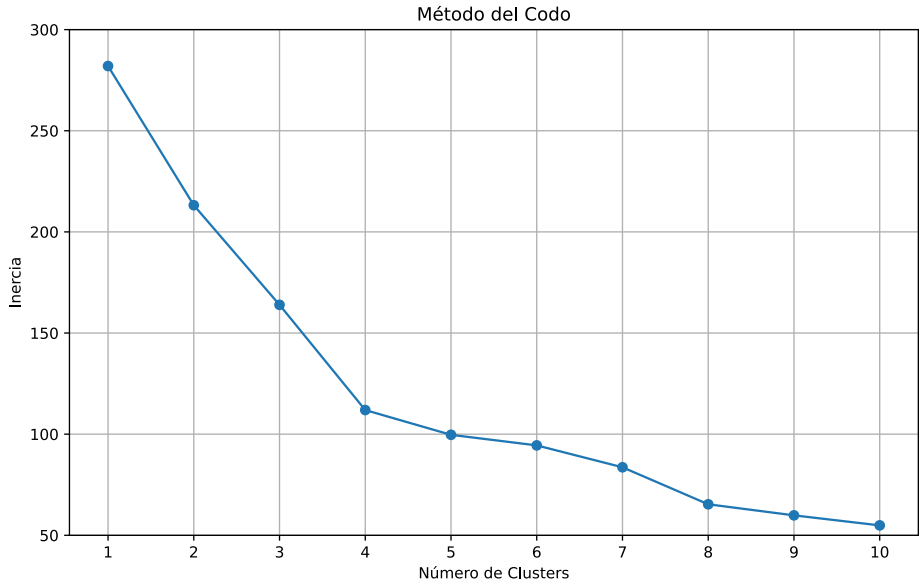


Figura 32. Representación gráfica del método del codo

3.2.2. Índice de Calinski-Harabasz

El índice de Calinski-Harabasz, al igual que el método del codo, es otra técnica utilizada para evaluar la calidad de un *clustering*. Se calcula como la razón entre la suma del cuadrado de las distancias entre centroides del *clusters* y la suma del cuadrado de las distancias intra-*cluster*. Un valor más alto de este indicador sugiere un *clustering* más compacto y mejor definido.

Por ejemplo, en la Figura 33, se puede observar que el valor más alto se obtiene para $k = 4$.

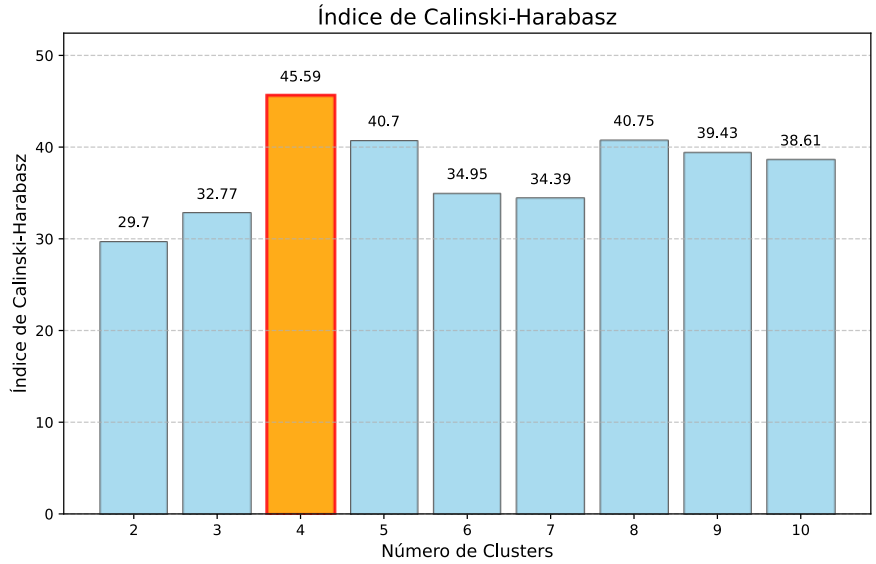


Figura 33. Índice de Calinski-Harabasz para distinto número de *clusters*.

3.2.1. Coeficiente de Silhouette

El coeficiente de Silhouette es una métrica que mide la calidad de un *clustering* basándose en cómo de similar es un punto a su propio *cluster* en comparación con otros *clusters*. Para cada punto, se calcula un valor de Silhouette que oscila entre -1 y 1. Este valor se obtiene a partir de las distancias medias intra-cluster (d_{intra}) e inter-cluster (d_{inter}) (Ecuación (6)).

$$S = \frac{1}{n_i} \sum \frac{d_{inter} - d_{intra}}{\max\left(d_{inter}, d_{intra}\right)} \tag{6}$$

El valor del coeficiente de Silhouette para todo el conjunto de datos se obtiene hallando el valor medio de S para todos los clústeres. A continuación, en la Figura 34, se muestran los distintos coeficientes de Silhouette obtenidos para distinto número de *clusters*.

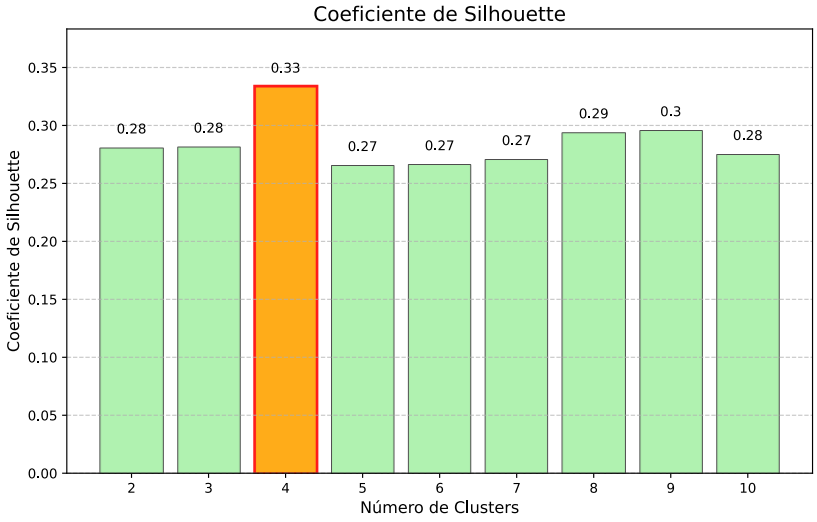


Figura 34. Coeficiente de Silhouette para distinto número de clusters.

Tal y como se puede observar en las tres figuras, todos los métodos sugieren un número óptimo de *clusters* igual a cuatro. A continuación, en la Figura 35, se muestra el resultado visual de la agrupación realizada y en la Figura 37 la media y desviación típica de las tres *features*.

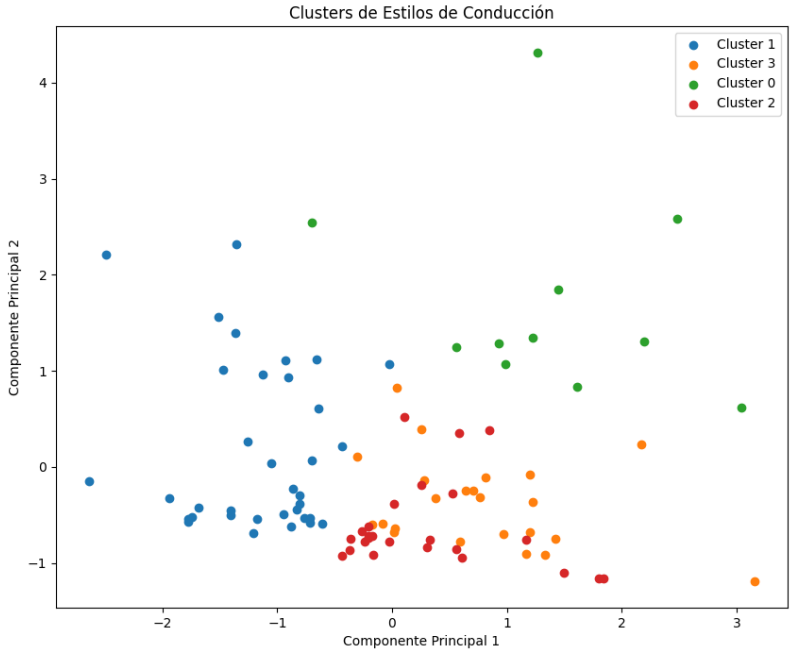


Figura 35. Representación gráfica de los cuatro clusters (k-Means).

A priori, podría parecer que la agrupación de *clusters* no es óptima, ya que el *Cluster* 2 (rojo) y el *Cluster* 3 (naranja) están demasiado próximos entre sí. Sin embargo, esta percepción se debe a la pérdida de información al representar los datos en un espacio bidimensional. En la Figura 36, se presenta la misma agrupación visualizada en un espacio tridimensional (3D) con tres componentes principales, donde la separación entre ambos *clusters* se aprecia de manera mucho más clara.

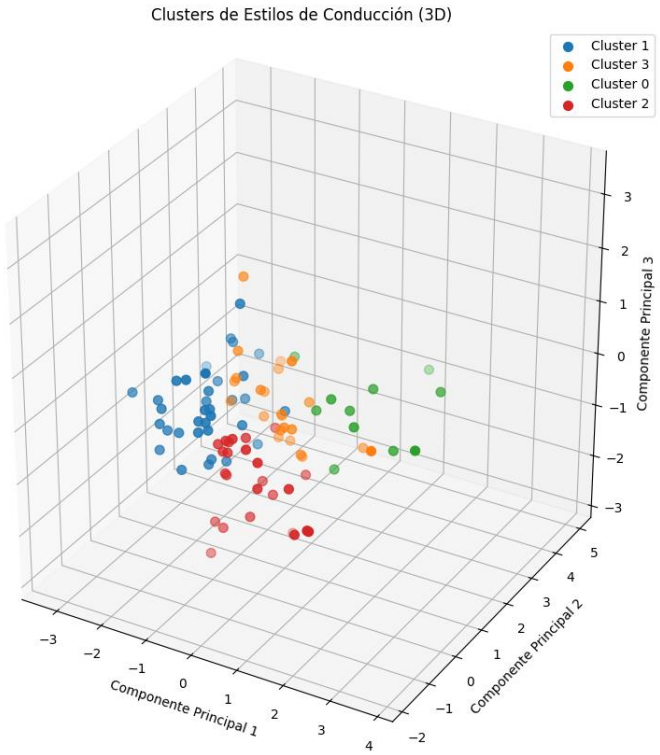


Figura 36. Representación gráfica 3D de los cuatro *clusters* (k-Means).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.42794 ± 0.13106	0.32442 ± 0.11466	0.02219 ± 0.00698
1	0.26755 ± 0.08011	0.06391 ± 0.07949	0.01749 ± 0.00746
2	0.55344 ± 0.09794	0.0729 ± 0.07241	0.01473 ± 0.00688
3	0.4284 ± 0.11618	0.02171 ± 0.04447	0.04484 ± 0.01114

Figura 37. Media y desviación típica de las tres *features* en cada uno de los *clusters* (k-Means).

A diferencia de lo que sucedía en Figura 29 donde simplemente se hacía una distinción entre conductor prudente e imprudente, en este caso también se subdividen los conductores imprudentes en tres perfiles distintos; agresivo, temerario y distraído; en función de si se supera la velocidad máxima permitida en la vía, en función del número de aceleraciones bruscas que se realizan y en función del uso del teléfono móvil, respectivamente.

Por ejemplo, los conductores del *cluster* 1 son los que menos tiempo se pasan del límite de velocidad y, como prácticamente no utilizan el móvil ni realizan aceleraciones bruscas, se podría considerar como el comportamiento más normal, siendo ligeramente agresivo.

En cambio, los usuarios del *cluster* 2 tienen un comportamiento notablemente más agresivo. Si bien es cierto que no utilizan demasiado el móvil ni realizan aceleraciones bruscas, son los que más tiempo superan la máxima velocidad permitida en la vía.

El *cluster* 3 está un paso por encima de este último grupo en términos de agresividad, llegando a representar un grupo de conductores temerarios. Es el segundo grupo que más se pasa del límite de velocidad y el grupo que más aceleraciones bruscas realiza. Esto sólo sucede si se producen frenazos, acelerones o giros mucho más fuertes de lo normal, lo que podría indicar que en alguna ocasión ha estado cerca de producirse un accidente. Sin embargo, cabe destacar que es, con diferencia, el grupo que menos utiliza el móvil. Esto puede parecer lógico puesto que, cuando un usuario supera el límite de velocidad es más improbable que utilice el móvil que un usuario que va a una velocidad moderada, al tener un menor control sobre el coche y el entorno. Lógicamente en ambos casos es un comportamiento de riesgo, pero en el caso de superar el límite máximo permitido más si cabe.

Por último, el *cluster* 0 indica un grupo distraído. Se supera la velocidad el 40% del tiempo, es el grupo que más utiliza el móvil y el segundo grupo que más aceleraciones bruscas realiza.

Tras hacer este primer análisis, se realiza exactamente el mismo procedimiento con el resto de técnicas de *clustering* obteniendo los resultados mostrados en la Tabla 2, donde CH el índice de Calinski-Harabasz y S es el coeficiente de Silhouette. Entre paréntesis aparece el número óptimo de *clusters* para estos métodos.

Tabla 2. Resumen del número óptimo de *clusters* y coeficiente de Silhouette obtenido por cada técnica de *clustering*.

Clustering	Codo (*k*)	CH (*k*)	S (*k*)	Perfiles de conducción
k-Means	121.22 (4)	45.59 (4)	0.33 (4)	- Prudente - Agresivo - Temerario - Distraído
Clustering Jerárquico	122.34 (2)	34.09 (2)	0.27 (2)	- Prudente - Imprudente
Clustering Espectral	132.35 (4)	28.32 (2)	0.35 (2)	- Prudente - Imprudente
Fuzzy C-Means	264.33 (2)	40.03 (2)	0.28 (2)	- Prudente - Imprudente
k-Medoids	120.02 (2)	39.47 (2)	0.28 (2)	- Prudente - Imprudente

El mejor coeficiente de Silhouette se obtiene con dos *clusters* utilizando el método de *Clustering* Espectral, obteniendo unas medias y desviaciones muy similares a las mostradas en la Figura 30.

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.5487 ± 0.12439	0.23643 ± 0.17118	0.32544 ± 0.04221
1	0.35634 ± 0.14035	0.06812 ± 0.08749	0.23676 ± 0.05707

Figura 38. Media y desviación típica de las tres *features* en cada uno de los *clusters* (*Clustering* Espectral).

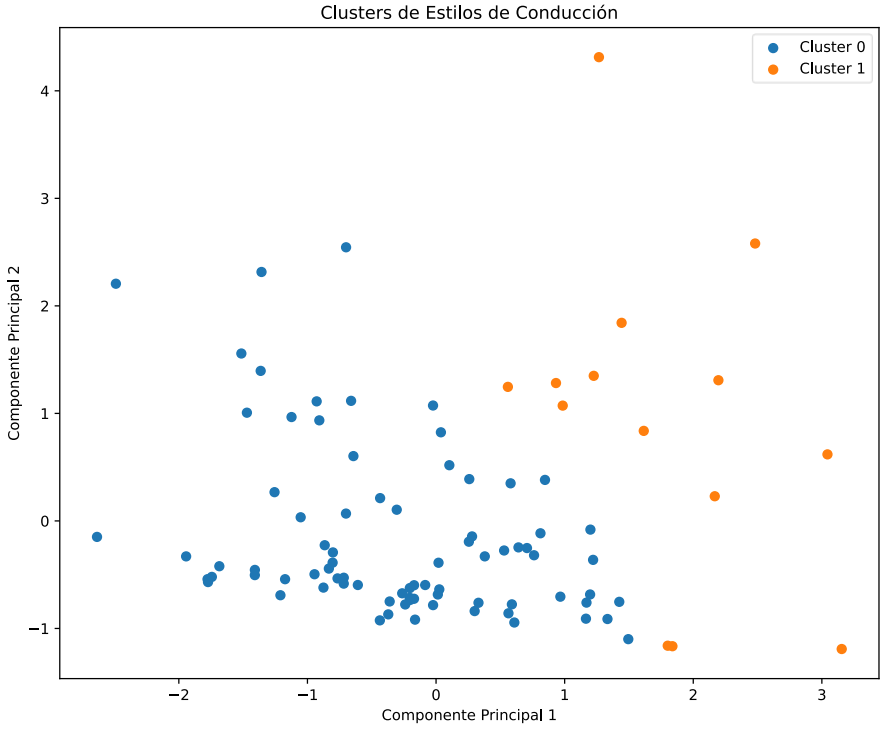


Figura 39. Representación gráfica de los dos *clusters* (*Clustering* espectral).

En defintiva, tal y como se ha visto en la Tabla 2, aunque la mayoría de métodos sugieren un número óptimo de *clusters* igual a dos, hay varios métodos que sugieren que el número óptimo de *clusters* también podría ser cuatro. En ese caso, se hace una distinción un poco más precisa donde se puede categorizar a los conductores como prudentes e imprudentes, y dentro de los conductores imprudentes a su vez categorizarlos en tres perfiles distintos; agresivo, temerario y distraído.

3.3. Data Augmentation

El aumento de datos, también conocido como *data augmentation*, es una estrategia que introduce variabilidad adicional en los datos sin necesidad de recopilar información nueva, lo cual es particularmente útil en contextos donde la obtención de datos adicionales es costosa o inviable. En este

trabajo, se ha implementado una técnica basada en ventanas temporales deslizantes para generar un número mayor de muestras de viajes. Específicamente, se crean ventanas de 5 minutos de duración con un desplazamiento de 1 minuto entre cada una. Este enfoque permite capturar diferentes segmentos temporales de un mismo viaje, aportando diversidad temporal al conjunto de datos.

La utilización de ventanas deslizantes ofrece varias ventajas. Por un lado, aumenta la densidad de datos disponibles para el entrenamiento de los modelos, lo que puede conducir a un mejor rendimiento y robustez frente a variaciones en los datos. Por otro lado, permite analizar el comportamiento y las características del viaje en intervalos más pequeños, facilitando la detección de patrones locales y eventos específicos que podrían pasar desapercibidos en análisis a mayor escala temporal. En la Figura 40, se muestra el procedimiento seguido para llevar a cabo el aumento de datos mediante ventanas deslizantes.

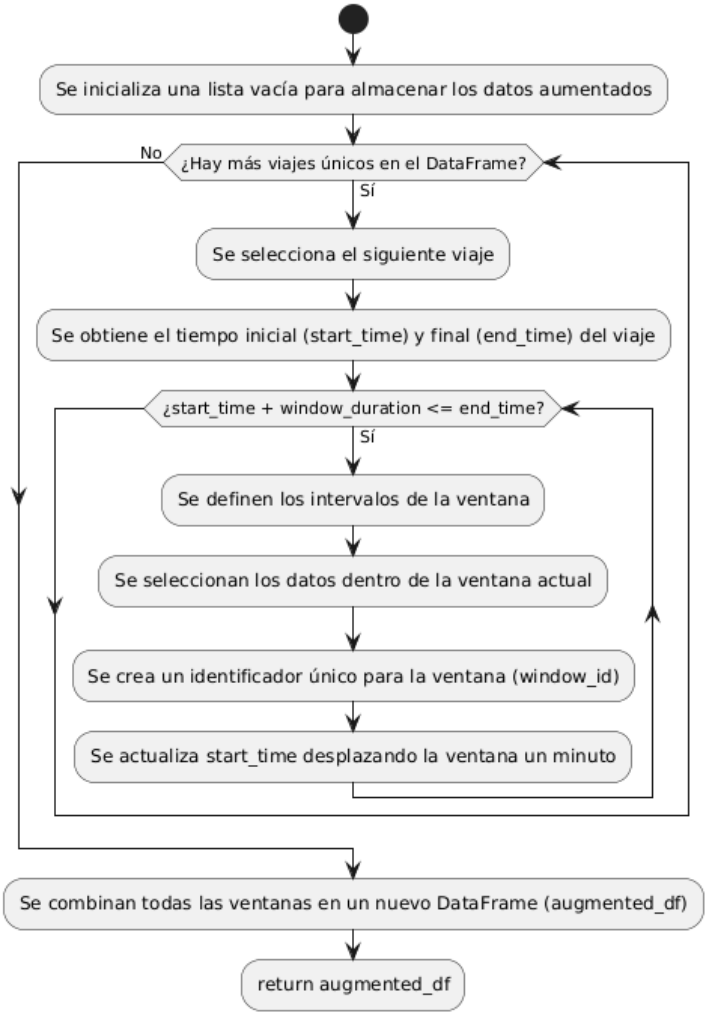


Figura 40. Función utilizada para el aumento de datos.

Tras aplicar este algoritmo al viaje mostrado en la Figura 15, se obtiene la Figura 41, donde se muestran únicamente 5 ventanas del total de ventanas generado en ese viaje.

Figura 41. Ejemplo de enventanado para el viaje 1652518433373+34555555555.

Con este procedimiento la muestra de viajes se ha visto aumentada notablemente (Figura 42).

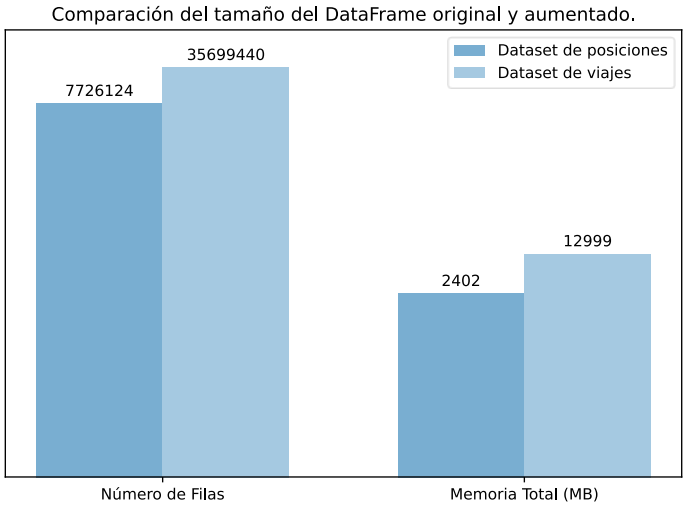


Figura 42. Comparación del tamaño del *DataFrame* original y aumentado en escala logarítmica.

Tras incrementar la variabilidad de los datos, se aplican nuevamente los algoritmos de *clustering* descritos en la sección anterior. El objetivo es evaluar si la ampliación de los datos genera diferencias significativas en los resultados, así como determinar su impacto en la calidad y robustez de las agrupaciones obtenidas. A continuación, en la Tabla 3, se muestran los resultados obtenidos.

Tabla 3. Resumen del número óptimo de *clusters* y coeficiente de Silhouette obtenido por cada técnica de *clustering*.

Clustering	**Codo (*k*)**	**CH (*k*)**	**S (*k*)**	**S (k) (antes del data augmentation)**	**Perfiles de conducción**
k-Means	109.34 (4)	47.23(4)	0.36 (5)	0.33 (4)	- Prudente - Agresivo - Temerario - Distraído - Fatigado
Clustering Jerárquico	90.07 (4)	45.34 (4)	0.35 (5)	0.26 (2)	- Prudente - Agresivo - Temerario - Distraído - Fatigado
Clustering Espectral	111.38 (4)	44.62 (4)	0.36 (4)	0.35 (2)	- Prudente - Agresivo - Temerario - Distraído
Fuzzy C-Means	247.12 (4)	46.64 (4)	0.34 (4)	0.28 (2)	- Prudente - Agresivo - Temerario - Distraído
k-Medoids	83.13 (5)	37.06 (5)	0.25 (5)	0.28 (2)	- Prudente - Agresivo - Temerario - Distraído - Fatigado

El análisis realizado muestra que, al igual que sucedía antes de realizar el aumento de datos, el valor más alto se obtiene con dos *clusters* utilizando el método de Clustering Espectral. Sin embargo, es importante destacar que se ha identificado una agrupación más específica de 4 ó 5 *clusters* con el resto de métodos (Figura 43), incrementando el valor del coeficiente de Silhouette.

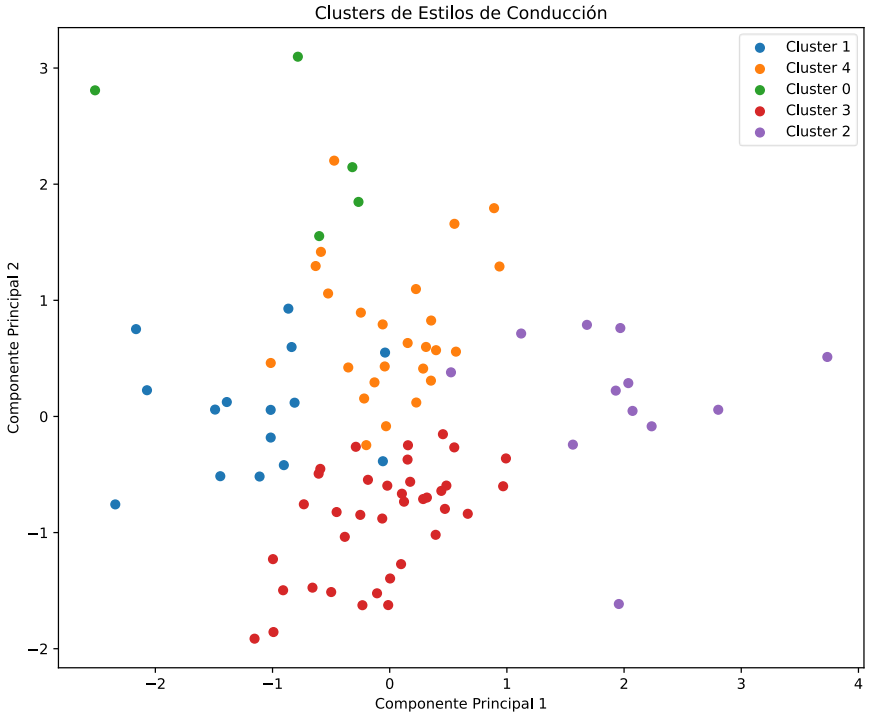


Figura 43. Representación gráfica de los cinco *clusters* (k-Means).

Se puede observar que los modelos mantienen un comportamiento similar antes y después del *data augmentation*, aunque con una ligera mejora en los resultados. Además, aparece la identificación de nuevos subgrupos de conductores, como un posible grupo de conductores fatigados, caracterizados por mantener velocidades normales y un uso prácticamente inexistente del teléfono móvil, pero con aceleraciones bruscas. Esto contrasta con los conductores distraídos, cuyas aceleraciones bruscas suelen estar asociadas a un uso excesivo del teléfono móvil.

En definitiva, se ha comprobado que el enriquecimiento de los datos no altera de manera significativa la estructura intrínseca de los viajes clasificados. Sin embargo, este proceso aporta una mayor robustez a las agrupaciones, así como una mejor cohesión interna, facilitando una caracterización más detallada de los distintos patrones de conducción existentes.

4. CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS

4.1. Conclusiones

El presente trabajo ha abordado la caracterización de estilos de conducción utilizando técnicas de aprendizaje automático aplicadas a un amplio conjunto de datos recopilados mediante una aplicación móvil. A continuación, se resumen las principales conclusiones obtenidas.

Las técnicas de *clustering* empleadas, como k-Means y Clustering Espectral, han sido de gran ayuda en la identificación de diferentes estilos de conducción. Se ha logrado categorizar a los conductores en distintos perfiles, incluyendo conductores prudentes, agresivos, temerarios y distraídos.

Las *features* más relevantes para la clasificación de estilos de conducción han sido el porcentaje de tiempo excediendo la velocidad máxima permitida, el uso del teléfono móvil y las aceleraciones bruscas realizadas. Estas variables han permitido una segmentación relativamente clara de los distintos conductores y han proporcionado identificadores valiosos sobre sus comportamientos al volante.

El proceso de limpieza y normalización de datos ha sido crucial para asegurar la validez del análisis. La eliminación de datos duplicados, anomalías y valores erróneos mejoró significativamente la calidad del conjunto de datos, permitiendo obtener resultados más precisos y representativos.

Se han cumplido todos los objetivos marcados al inicio del proyecto, aunque aún se podría haber añadido un objetivo adicional para validar el comportamiento de los distintos modelos. Se podrían plantear dos opciones, detalladas a continuación.

- Realizar pruebas de conducción en un entorno controlado y seguro, simulando distintos comportamientos al volante para, posteriormente, verificar la correcta clasificación de nuestro modelo. Por un tema de plazos con la empresa externa con la que colabora Centro Zaragoza, finalmente no se ha podido incluir esta validación en el presente trabajo.
- Buscar *datasets* etiquetados de un problema parecido al expuesto en este trabajo y que, además, tuviesen un formato similar al de los datos que se tienen. Se ha

buscado en los principales portales web [34], [35] y, si bien es cierto que se han encontrado *datasets* de problemas como el que se ha tratado en este trabajo, no aparecían todas las *features*. Algunos podían incluir las aceleraciones, pero no el uso del móvil o la velocidad máxima permitida en la vía [36], [37], [38].

Los resultados de este estudio pueden tener aplicaciones prácticas en la mejora de los sistemas de asistencia al conductor, contribuyendo a la personalización de los vehículos y a la promoción de estilos de conducción más seguros y eficientes. Además, pueden ser útiles para las aseguradoras en la personalización de primas y para los urbanistas en la planificación de infraestructuras viales.

4.2. Líneas de investigación futuras

El trabajo realizado abre diversas líneas de investigación y desarrollo futuras:

- Ampliación del conjunto de datos: Para mejorar la generalización de los modelos desarrollados sería beneficioso ampliar el conjunto de datos incluyendo un mayor número de conductores y viajes, además de añadir datos de otras regiones del mundo que, junto con la incorporación de las condiciones de tráfico, podría enriquecer el análisis actual.
- Mejora de algoritmos: Aunque los métodos de *clustering* utilizados proporcionaron buenos resultados, explorar otras técnicas más avanzadas, como redes neuronales profundas y algoritmos de aprendizaje por refuerzo, podría mejorar aún más la precisión y robustez de la clasificación. Se podría probar con Deep Embedded Clustering (DEC), un método específico que combina *autoencoders* y una técnica de *clustering* en un solo proceso [39]; o ClusterGAN, donde la red generativa intenta producir datos que puedan ser fácilmente clusterizados por la red discriminativa, lo que permite generar grupos naturales dentro de los datos [40].
- Creación de nuevas *features*: En este trabajo, solo se ha considerado el porcentaje de tiempo que un conductor excede la velocidad máxima permitida en la vía. Sin embargo, no es lo mismo exceder la velocidad por 5 km/h que por 50 km/h. Por tanto, sería útil incluir una característica que mida la media de la cantidad por la cual se ha superado la velocidad máxima a lo largo del viaje. Aparte de esta *feature*, se podrían añadir más umbrales en la aceleración, como un umbral ECO,

que permita identificar a los conductores más responsables y comprometidos con el medio ambiente.

- Integración en tiempo real: Desarrollar sistemas capaces de analizar y clasificar estilos de conducción en tiempo real podría tener aplicaciones inmediatas en vehículos autónomos y sistemas de asistencia al conductor. Esto requeriría optimizar los algoritmos para su ejecución en dispositivos con recursos limitados. Además, de esta forma, se podrían realizar estudios longitudinales para evaluar el impacto de la personalización de los sistemas de asistencia al conductor basada en los estilos de conducción identificados. Esto permitiría medir mejoras en la seguridad vial y la eficiencia energética.
- Identificación de nuevos patrones de conducción: Se podrían explorar otras aproximaciones en la caracterización de estilos de conducción, como la identificación de conductores *eco-friendly* o conductores que están bajo la influencia de sustancias.
- Correlación de agresividad en redes sociales y carretera: Siempre que el usuario registrado vinculase la aplicación con sus redes sociales, se podrían utilizar modelos de Deep Learning basados en Procesamiento de Lenguaje Natural (NLP) para clasificar su agresividad en redes sociales e intentar buscar algún tipo de correlación con su agresividad al volante.

4.3. Posibles aplicaciones

El análisis detallado y la caracterización de los estilos de conducción ofrecen un vasto campo de aplicaciones prácticas en diversas áreas. A continuación, se presentan algunas de las posibles aplicaciones más relevantes.

- Servicios personalizados en aseguradoras: Se podrían desarrollar programas de incentivos por buen comportamiento al volante, además de incluir servicios de asistencia en carretera en caso de accidente. Es decir, si se monitorizasen los datos de los asegurados en tiempo real y se detectasen comportamientos anómalos (por ejemplo, llevar 5 minutos parado en un tramo de 120 Km/h), se podría llamar al asegurado y si no responde enviar inmediatamente una ambulancia para asistirle, porque lo más probable es que haya sufrido un accidente.

- Gestión de reclamaciones: Los datos pueden ayudar a determinar la velocidad del vehículo, la localización y las condiciones del incidente, facilitando la resolución de disputas.

5. REFERENCIAS

[1] Python Software Foundation, “Python,” https://www.python.org.

[2] Inc. Anaconda, “Anaconda,” https://www.anaconda.com.

[3] Project Jupyter, “Jupyter,” https://jupyter.org.

[4] Spyder, “Spyder – The Scientific Python Development Environment,” https://www.spyder-ide.org.

[5] Software Freedom Conservancy, “Git,” https://git-scm.com.

[6] S. Hemminki, P. Nurmi, and S. Tarkoma, “Accelerometer-based transportation mode detection on smartphones,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, New York, NY, USA: ACM, Nov. 2013, pp. 1–14. doi: 10.1145/2517351.2517367.

[7] D. Shin *et al.*, “Urban sensing: Using smartphones for transportation mode classification,” *Comput Environ Urban Syst*, vol. 53, pp. 76–86, Sep. 2015, doi: 10.1016/j.compenvurbsys.2014.07.011.

[8] A. Ghose, A. Chowdhury, V. Chandel, T. Banerjee, and T. Chakravarty, “An enhanced automated system for evaluating harsh driving using smartphone sensors,” in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, New York, NY, USA: ACM, Jan. 2016, pp. 1–6. doi: 10.1145/2833312.2849555.

[9] A. Chowdhury, T. Chakravarty, A. Ghose, T. Banerjee, and P. Balamuralidhar, “Investigations on Driver Unique Identification from Smartphone’s GPS Data Alone,” *J Adv Transp*, vol. 2018, pp. 1–11, 2018, doi: 10.1155/2018/9702730.

[10] A. Noureldin, A. El-Shafie, and M. Bayoumi, “GPS/INS integration utilizing dynamic neural networks for vehicular navigation,” *Information Fusion*, vol. 12, no. 1, pp. 48–57, Jan. 2011, doi: 10.1016/j.inffus.2010.01.003.

[11] J. Wahlstrom, I. Skog, and P. Handel, “Detection of Dangerous Cornering in GNSS-Data-Driven Insurance Telematics,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3073–3083, Dec. 2015, doi: 10.1109/TITS.2015.2431293.

[12] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, “Estimating driving behavior by a smartphone,” in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2012, pp. 234–239. doi: 10.1109/IVS.2012.6232298.

[13] B. Bose, J. Dutta, S. Ghosh, P. Pramanick, and S. Roy, “D&RSense: Detection of Driving Patterns and Road Anomalies,” in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, IEEE, Feb. 2018, pp. 1–7. doi: 10.1109/IoT-SIU.2018.8519861.

[14] B. Predic and D. Stojanovic, “Enhancing driver situational awareness through crowd intelligence,” *Expert Syst Appl*, vol. 42, no. 11, pp. 4892–4909, Jul. 2015, doi: 10.1016/j.eswa.2015.02.013.

[15] E. G. Mantouka, E. N. Barmpounakis, and E. I. Vlahogianni, "Identifying driving safety profiles from smartphone data using unsupervised learning," *Saf Sci*, vol. 119, pp. 84–90, Nov. 2019, doi: 10.1016/j.ssci.2019.01.025.

[16] E. Papadimitriou, A. Argyropoulou, D. I. Tselentis, and G. Yannis, "Analysis of driver behaviour through smartphone data: The case of mobile phone use while driving," *Saf Sci*, vol. 119, pp. 91–97, Nov. 2019, doi: 10.1016/j.ssci.2019.05.059.

[17] K. Ben Ahmed, B. Goel, P. Bharti, S. Chellappan, and M. Bouhorma, "Leveraging Smartphone Sensors to Detect Distracted Driving Activities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3303–3312, Sep. 2019, doi: 10.1109/TITS.2018.2873972.

[18] C. Andrieu and G. Saint Pierre, "Using statistical models to characterize eco-driving style with an aggregated indicator," in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2012, pp. 63–68. doi: 10.1109/IVS.2012.6232197.

[19] G. Castignani, R. Frank, and T. Engel, "An evaluation study of driver profiling fuzzy algorithms using smartphones," in *2013 21st IEEE International Conference on Network Protocols (ICNP)*, IEEE, Oct. 2013, pp. 1–6. doi: 10.1109/ICNP.2013.6733681.

[20] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, "Wolverine: Traffic and road condition estimation using smartphone sensors," in *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, IEEE, Jan. 2012, pp. 1–6. doi: 10.1109/COMSNETS.2012.6151382.

[21] C. Saiprasert, T. Pholprasit, and S. Thajchayapong, "Detection of Driving Events using Sensory Data on Smartphone," *International Journal of Intelligent Transportation Systems Research*, vol. 15, no. 1, pp. 17–28, Jan. 2017, doi: 10.1007/s13177-015-0116-5.

[22] D.-W. Koh and H.-B. Kang, "Smartphone-based modeling and detection of aggressiveness reactions in senior drivers," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Jun. 2015, pp. 12–17. doi: 10.1109/IVS.2015.7225655.

[23] D. Yi, J. Su, C. Liu, M. Quddus, and W.-H. Chen, "A machine learning based personalized system for driving state recognition," *Transp Res Part C Emerg Technol*, vol. 105, pp. 241–261, Aug. 2019, doi: 10.1016/j.trc.2019.05.042.

[24] E. I. Vlahogianni and E. N. Barmpounakis, "Driving analytics using smartphones: Algorithms, comparisons and challenges," *Transp Res Part C Emerg Technol*, vol. 79, pp. 196–206, Jun. 2017, doi: 10.1016/j.trc.2017.03.014.

[25] J. E. Meseguer, C. K. Toh, C. T. Calafate, J. C. Cano, and P. Manzoni, "Drivingstyles: a mobile platform for driving styles and fuel consumption characterization," *Journal of Communications and Networks*, vol. 19, no. 2, pp. 162–168, Apr. 2017, doi: 10.1109/JCN.2017.000025.

[26] H. R. Eftekhari and M. Ghatee, "An inference engine for smartphones to preprocess data and detect stationary and transportation modes," *Transp Res Part C Emerg Technol*, vol. 69, pp. 313–327, Aug. 2016, doi: 10.1016/j.trc.2016.06.005.

[27] T. K. Chan, C. S. Chin, H. Chen, and X. Zhong, "A Comprehensive Review of Driver Behavior Analysis Utilizing Smartphones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4444–4475, Oct. 2020, doi: 10.1109/TITS.2019.2940481.

[28] R. C. Hibbeler, *Engineering Mechanics: Dynamics*, 14th ed. Pearson, 2016.

[29] J. L. Meriam and L. G. Kraige, *Engineering Mechanics: Dynamics*, 7th ed. Wiley, 2012.

[30] F. P. Beer, E. R. Johnston, and P. J. Cornwell, *Vector Mechanics for Engineers: Dynamics*, 11th ed. McGraw-Hill Education, 2015.

[31] O. Cisneros, "Perfiles de conductor esperados y parámetros a monitorizar," Mar. 2023.

[32] Stanford University, "K-Means Clustering," https://stanford.edu/~cpiech/cs221/handouts/kmeans.html.

[33] U. von Luxburg, "A Tutorial on Spectral Clustering," Nov. 2007.

[34] Kaggle, "Kaggle: Your Machine Learning and Data Science Community," https://www.kaggle.com.

[35] UCI Machine Learning Repository, "UCI Machine Learning Repository," https://archive.ics.uci.edu/ml.

[36] S. Work, "Driving Behavior Dataset ," Kaggle.

[37] V. Krishna, "Aggressive Driving Data," Kaggle.

[38] E. Karan, "Driver Behaviour Analysis Using Sensor," Kaggle.

[39] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," Nov. 2015.

[40] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, "ClusterGAN: Latent Space Clustering in Generative Adversarial Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4610–4617, Jul. 2019, doi: 10.1609/aaai.v33i01.33014610.

Universidad Zaragoza
1542

Trabajo Fin de Máster
ANEXOS

Clasificación de estilos de conducción utilizando técnicas de machine learning

Classification of driving behaviour using machine learning techniques

Autor

Luis Llorente Muro

Director

Julio David Buldain Pérez

Departamento de Ingeniería Electrónica y Comunicaciones

Escuela de Ingeniería y Arquitectura

2024-2025

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ANEXO A. CÓDIGO DESARROLLADO PARA DESCARGAR CSV

En este apartado se muestra el *script* desarrollado para descargar los archivos CSV con los que se ha trabajado a lo largo de este proyecto.

```
# -*- coding: utf-8 -*-
"""
Created on Tue Aug 20 08:26:57 2024

@author: L.Llorente
"""

import os
from os import listdir
from os.path import isfile, join
import webbrowser
from pathlib import Path
import shutil

folders = ['Positions', 'Travels']

# Se buscan todos los CSV del directorio de trabajo
for folder in folders:

    path = os.getcwd() + "\\" + folder
    lista_csv = [f for f in listdir(path) if isfile(join(path, f))]

    for csv in lista_csv:

        # Como los CSV tienen un formato especial que no se puede cargar
con Pandas se leen como un archivo de texto
        csv_path = path + "\\" + csv
        file = open(csv_path, "r")
        content = file.read()
        file.close()

        # Se busca entre dos caracteres el enlace de descarga del CSV con
formato válido
        if folder == "Positions":
            link_first_idx = [i for i, c in enumerate(content) if c ==
'œ'][-1]
        else:
            link_first_idx = [i for i, c in enumerate(content) if c ==
'š'][-1]
        link_second_idx = [i for i, c in enumerate(content) if c ==
'_'][-1]
        link = content[link_first_idx + 1 : link_second_idx]
```

```
        # Se abre el enlace en el explorador para descargar el CSV con
formato válido
        webbrowser.open(link)

        # Se copia el CSV descargado al directorio de trabajo. El CSV
descargado ya no empieza por "._"
        downloads_path = str(Path.home() / "Downloads") + "\\" + csv[2:]
        dst_path = os.getcwd() + "\\CSV\\"
        if not os.path.isdir(dst_path):
            os.mkdir(dst_path)

        # Hay veces que se ejecuta antes la acción de copiar que la
propia descarga en sí, por lo que se espera a que se haya descargado
        download_copy = None
        while download_copy is None:
            try:
                download_copy = shutil.copyfile(downloads_path, dst_path
+ "\\" + csv[2:])
            except:
                 pass
```

ANEXO B. RESULTADOS ANTES DE DATA AUGMENTATION

En este apartado se muestran las gráficas obtenidas con los distintos métodos utilizados para estimar el número óptimo de *clusters* (método del codo, índice de Calinski-Harabasz y coeficiente de Silhouette), así como una representación de los *clusters* obtenidos con los datos originales, sin haber utilizado técnicas de *data augmentation*.

B.1. k-Means

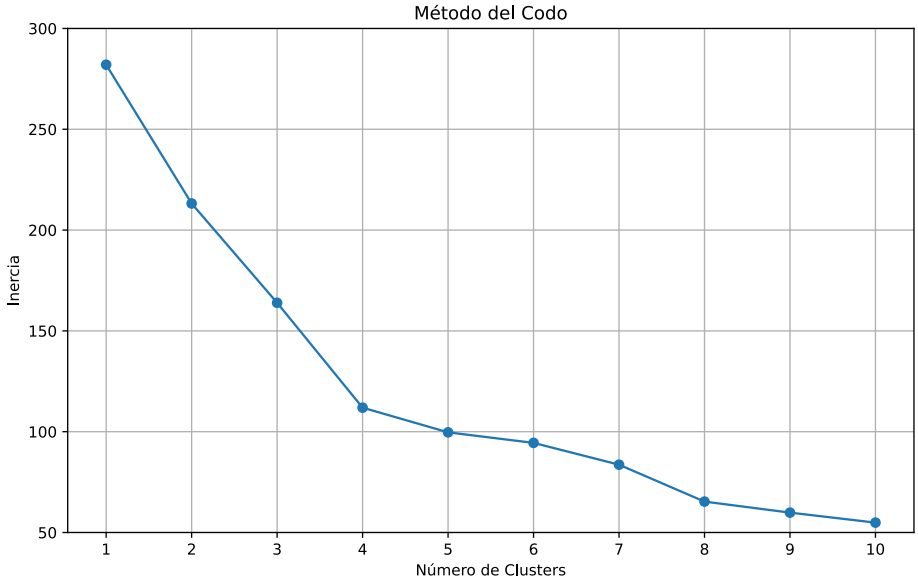


Figura 44. Representación gráfica del método del codo (k-Means).

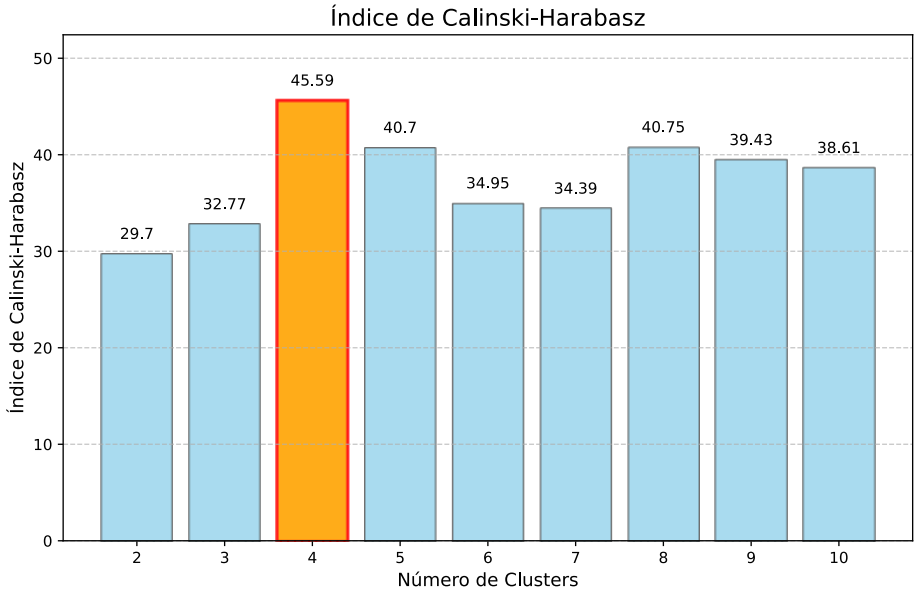


Figura 45. Índice de Calinski-Harabasz para distinto número de *clusters* (k-Means).

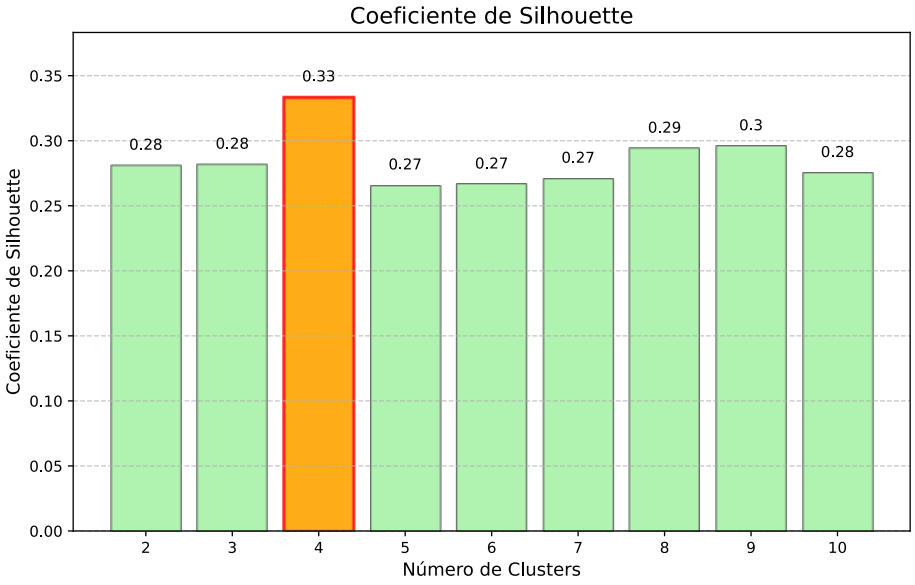


Figura 46. Coeficiente de Silhouette para distinto número de *clusters* (k-Means).

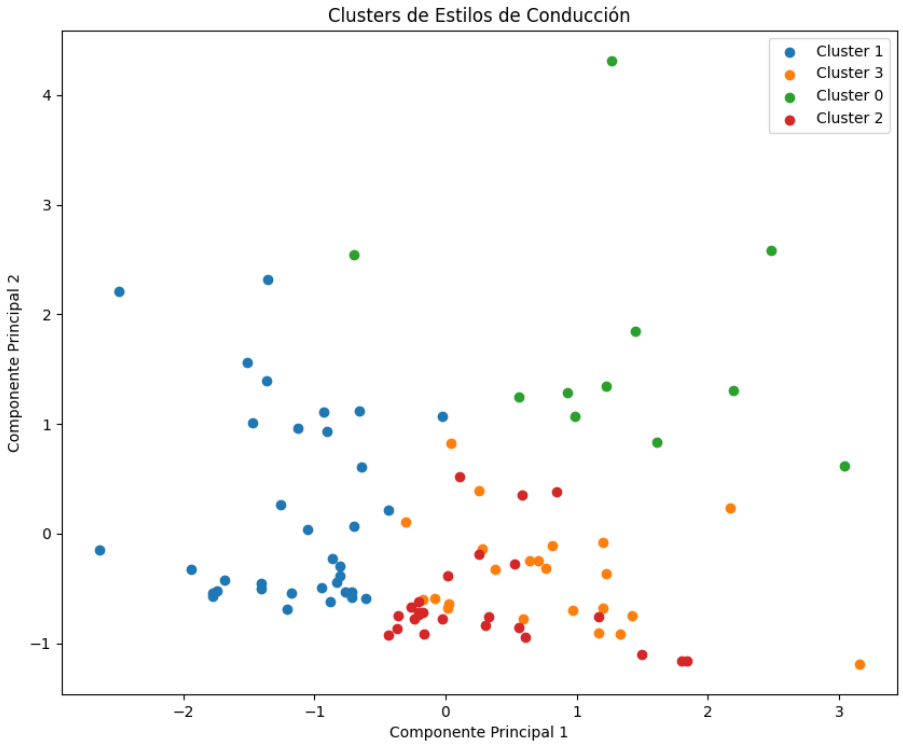


Figura 47. Representación gráfica de los *clusters* (k-Means).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.42794 ± 0.13106	0.32442 ± 0.11466	0.02219 ± 0.00698
1	0.26755 ± 0.08011	0.06391 ± 0.07949	0.01749 ± 0.00746
2	0.55344 ± 0.09794	0.0729 ± 0.07241	0.01473 ± 0.00688
3	0.4284 ± 0.11618	0.02171 ± 0.04447	0.04484 ± 0.01114

Figura 48. Media y desviación típica de las tres *features* en cada uno de los *clusters* (k-Means).

B.2. Clustering Espectral

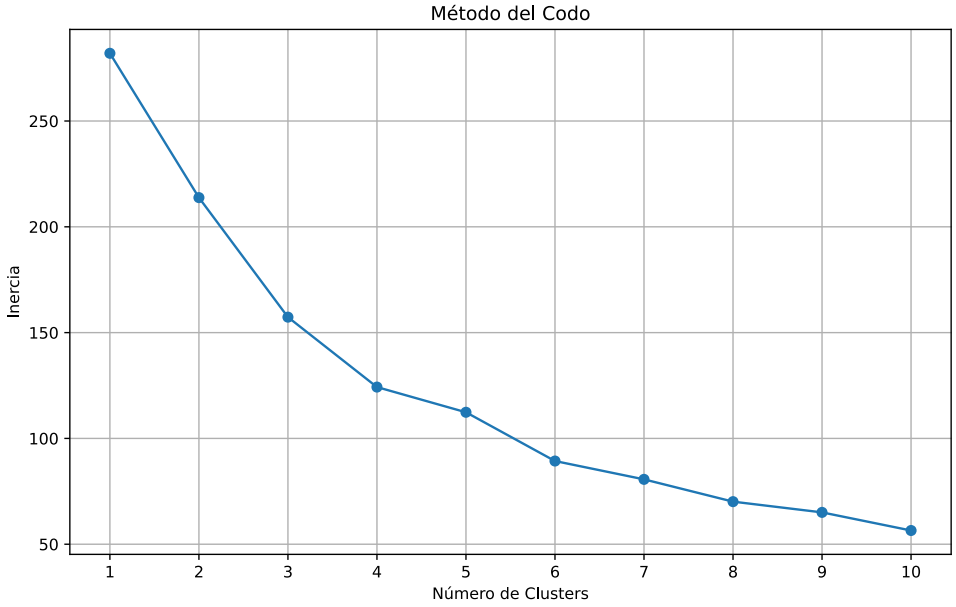


Figura 49. Representación gráfica del método del codo (Clustering Espectral).

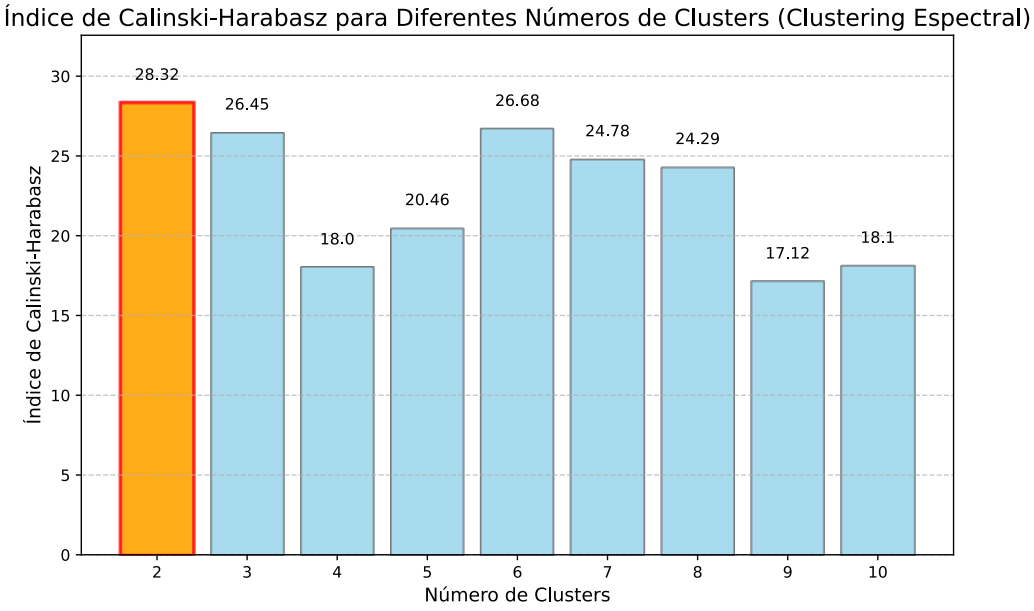


Figura 50. Índice de Calinski-Harabasz para distinto número de *clusters* (Clustering Espectral).

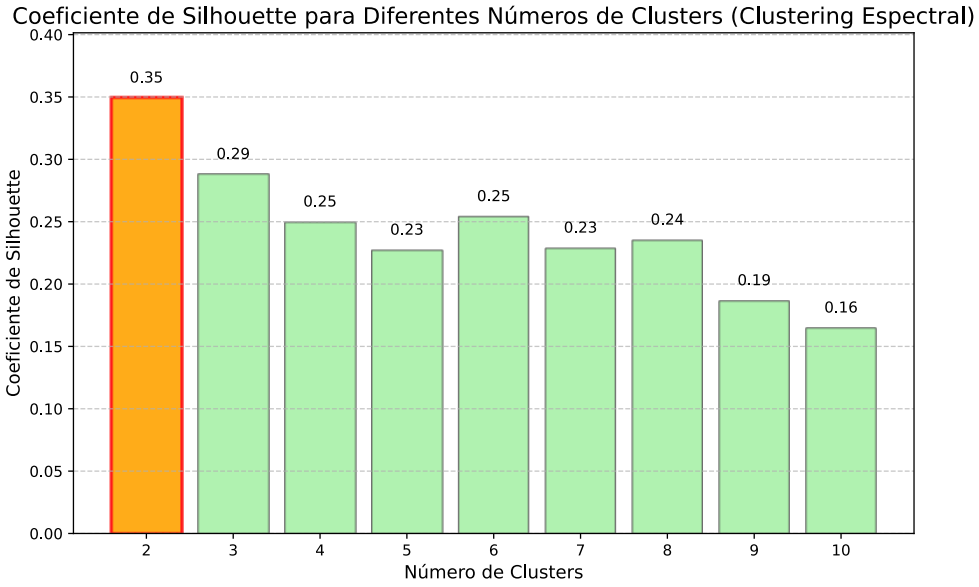


Figura 51. Coeficiente de Silhouette para distinto número de *clusters* (Clustering Espectral).

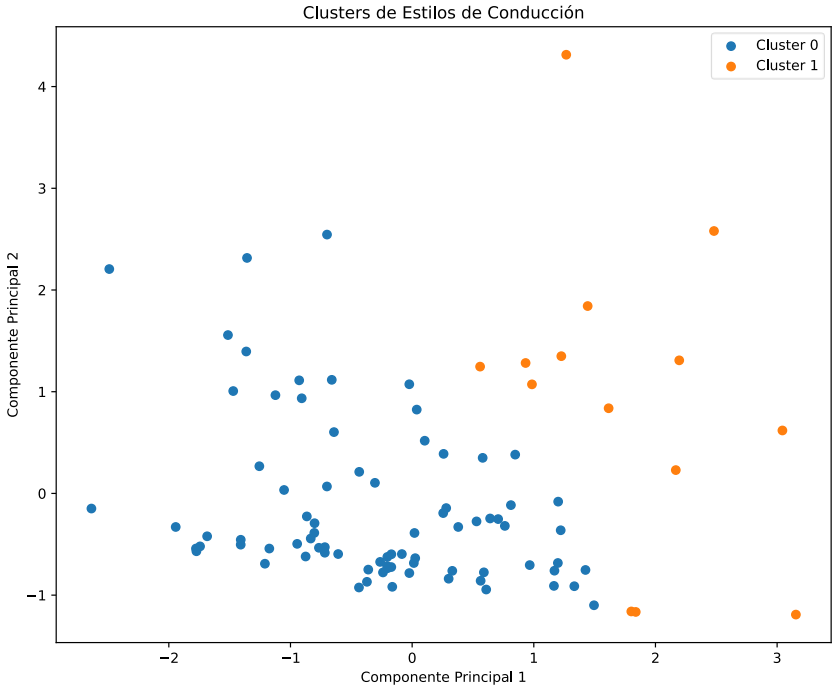


Figura 52. Representación gráfica de los *clusters* (Clustering Espectral).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.35634 ± 0.14035	0.06812 ± 0.08749	0.23676 ± 0.05707
1	0.5487 ± 0.12439	0.23643 ± 0.17118	0.32544 ± 0.04221

Figura 53. Media y desviación típica de las tres *features* en cada uno de los *clusters* (Clustering Espectral).

B.3. Clustering Jerárquico

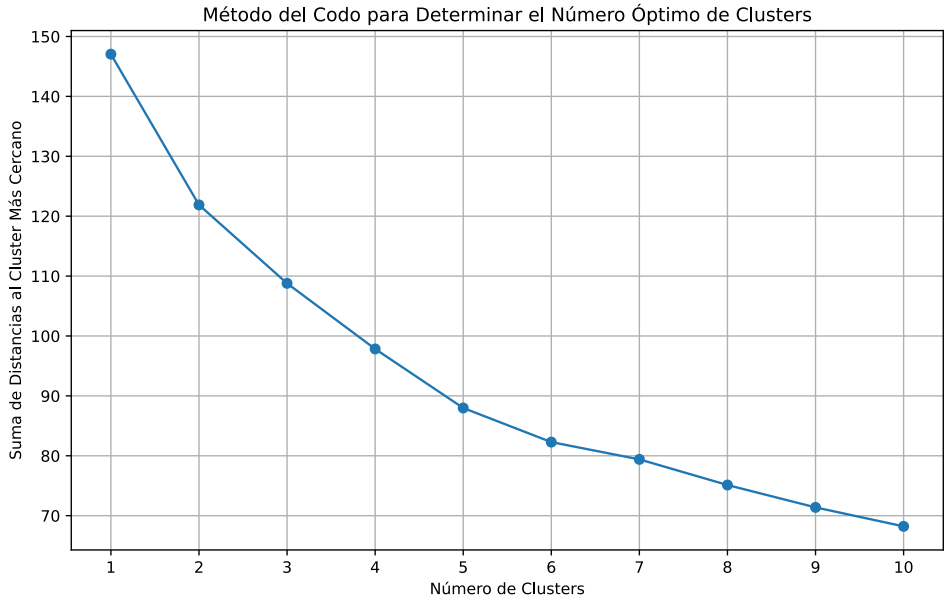


Figura 54. Representación gráfica del método del codo (Clustering Jerárquico).

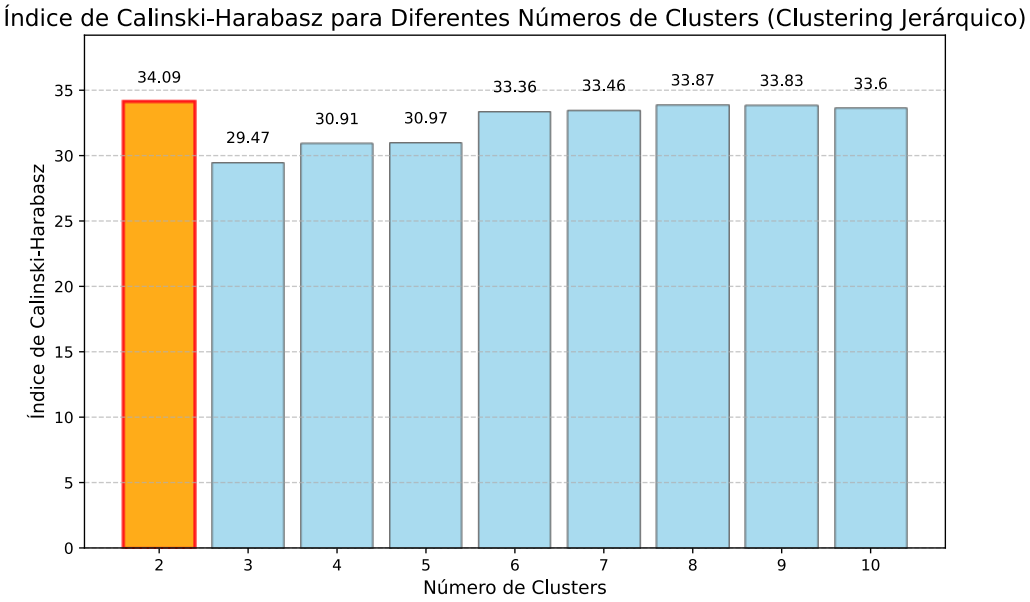


Figura 55. Índice de Calinski-Harabasz para distinto número de *clusters* (Clustering Jerárquico).

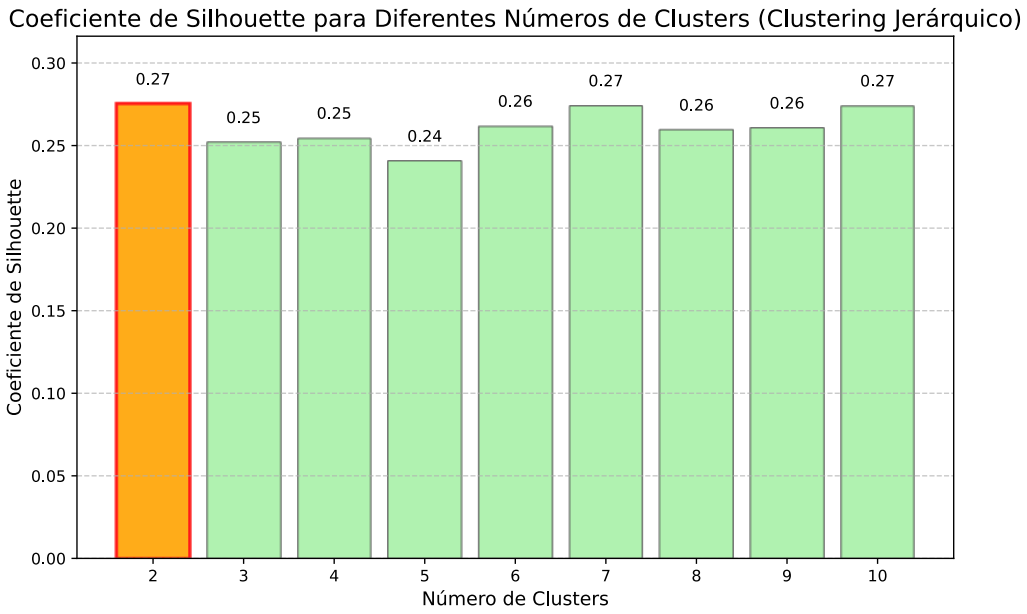


Figura 56. Coeficiente de Silhouette para distinto número de *clusters* (Clustering Jerárquico).

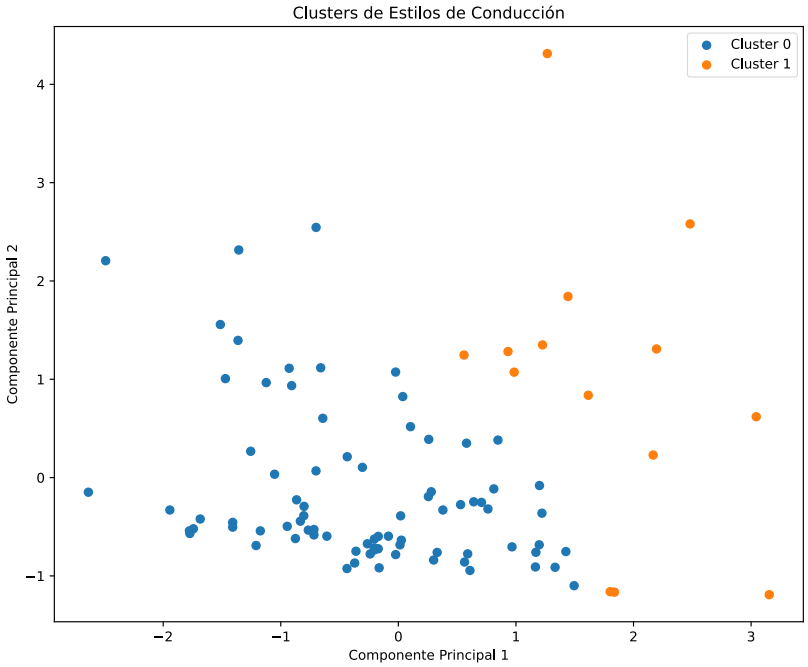


Figura 57. Representación gráfica de los *clusters* (Clustering Jerárquico).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.5487 ± 0.12439	0.23643 ± 0.17118	0.32544 ± 0.04221
1	0.35634 ± 0.14035	0.06812 ± 0.08749	0.23676 ± 0.05707

Figura 58. Media y desviación típica de las tres *features* en cada uno de los *clusters* (Clustering Jerárquico).

B.4. Fuzzy C-Means

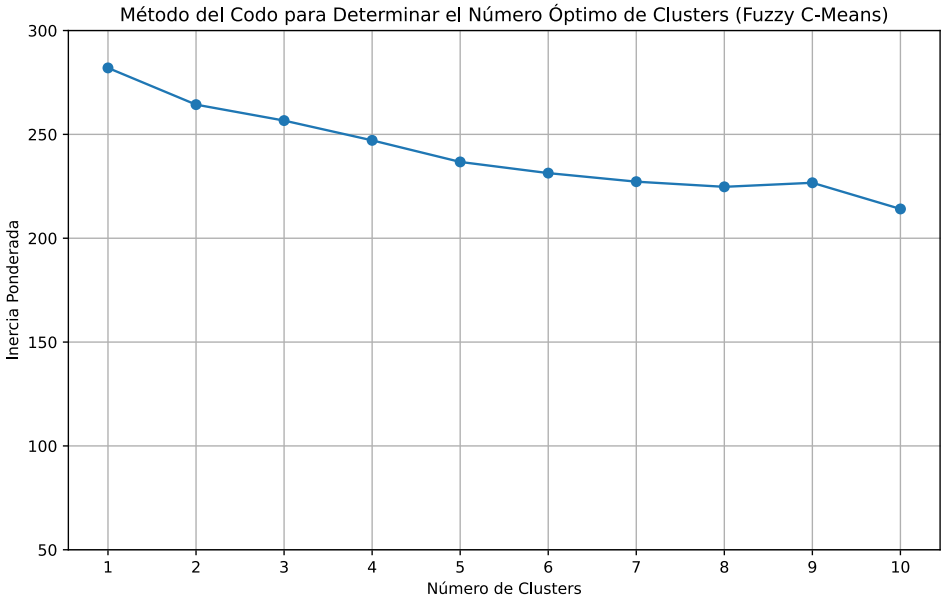


Figura 59. Representación gráfica del método del codo (Fuzzy C-Means).

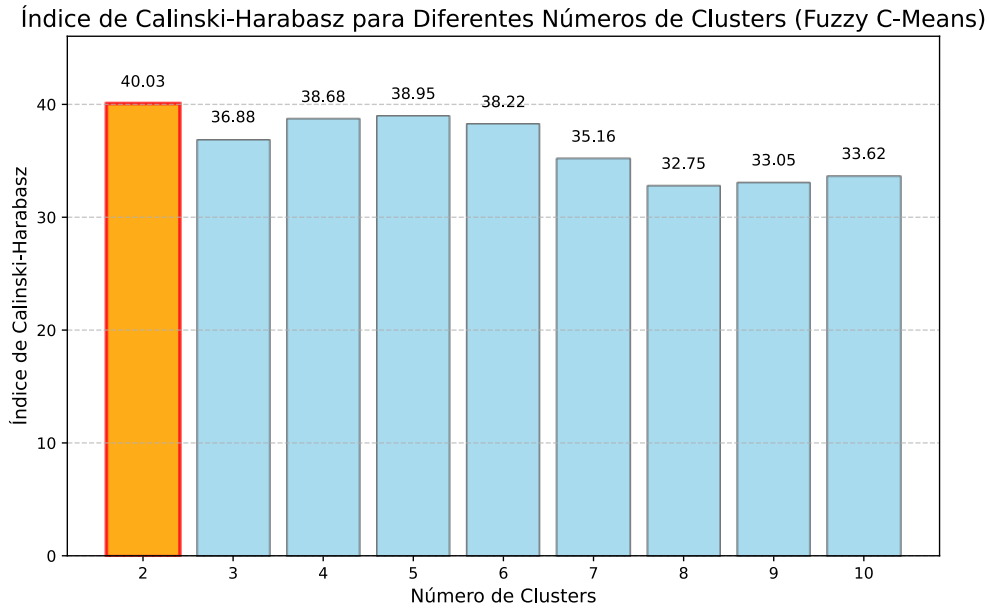


Figura 60. Índice de Calinski-Harabasz para distinto número de *clusters* (Fuzzy C-Means).

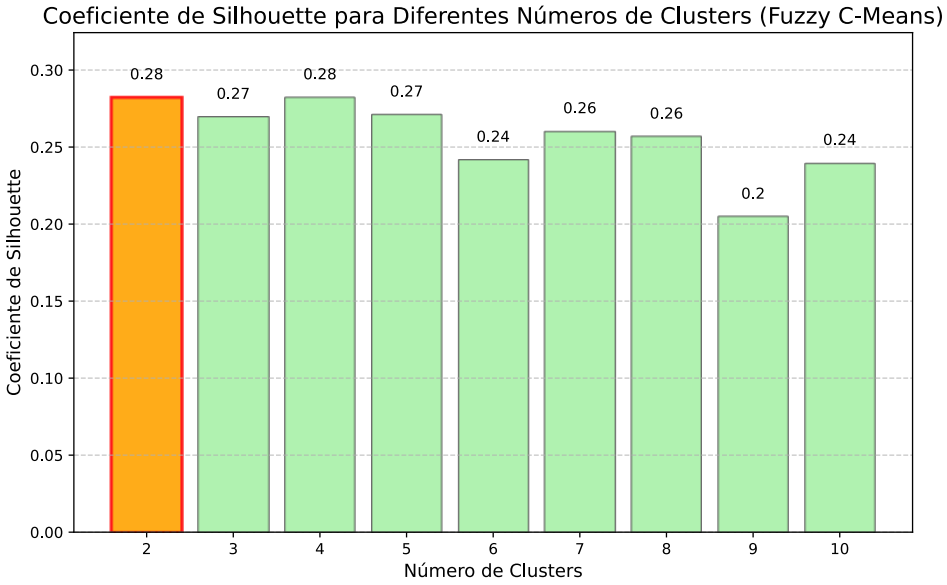


Figura 61. Coeficiente de Silhouette para distinto número de *clusters* (Fuzzy C-Means).

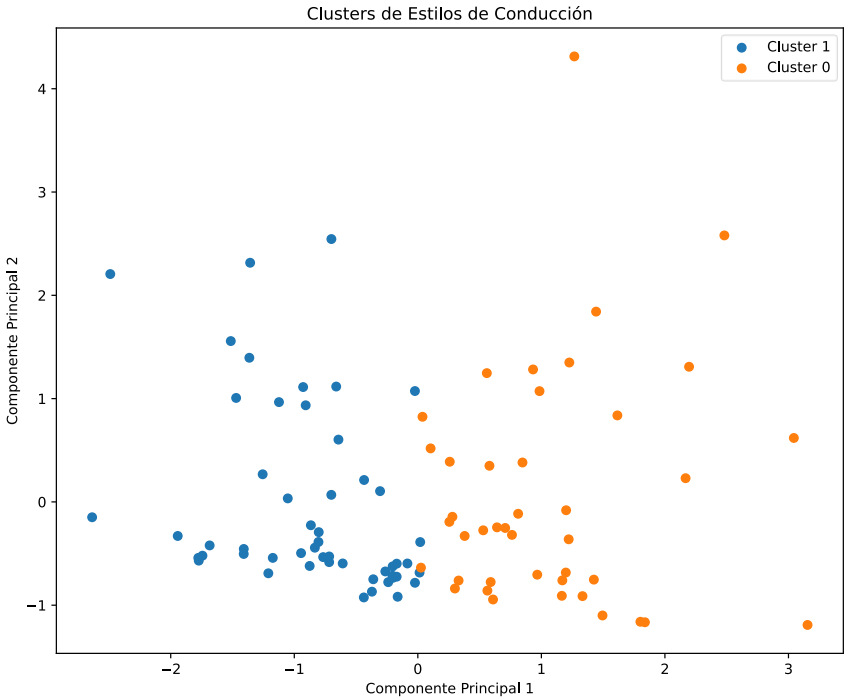


Figura 62. Representación gráfica de los *clusters* (Fuzzy C-Means).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.4996 ± 0.11879	0.11933 ± 0.13602	0.29529 ± 0.04932
1	0.29242 ± 0.11048	0.07208 ± 0.10027	0.21336 ± 0.0482

Figura 63. Media y desviación típica de las tres *features* en cada uno de los *clusters* (Fuzzy C-Means).

B.5. k-Medoides

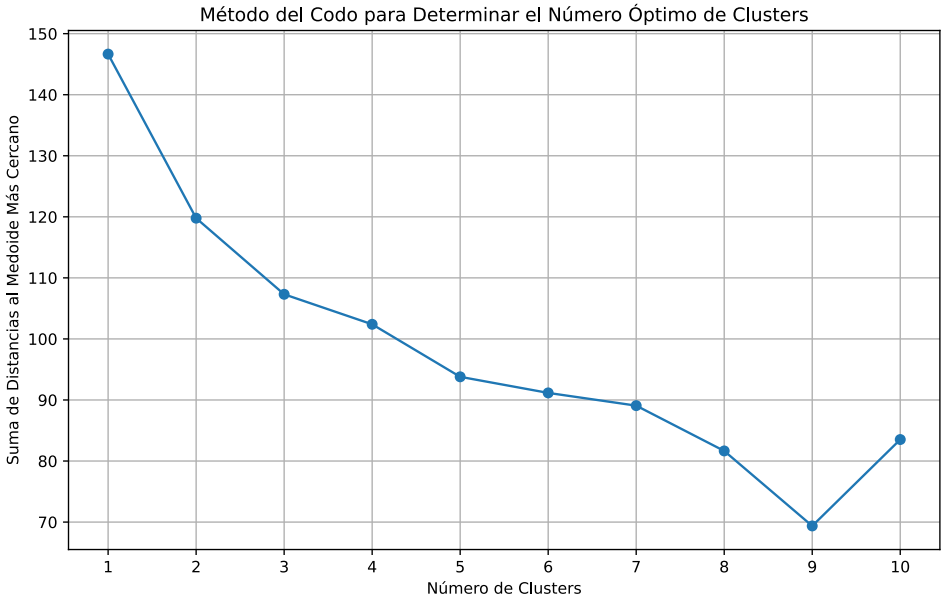


Figura 64. Representación gráfica del método del codo (k-Medoides).

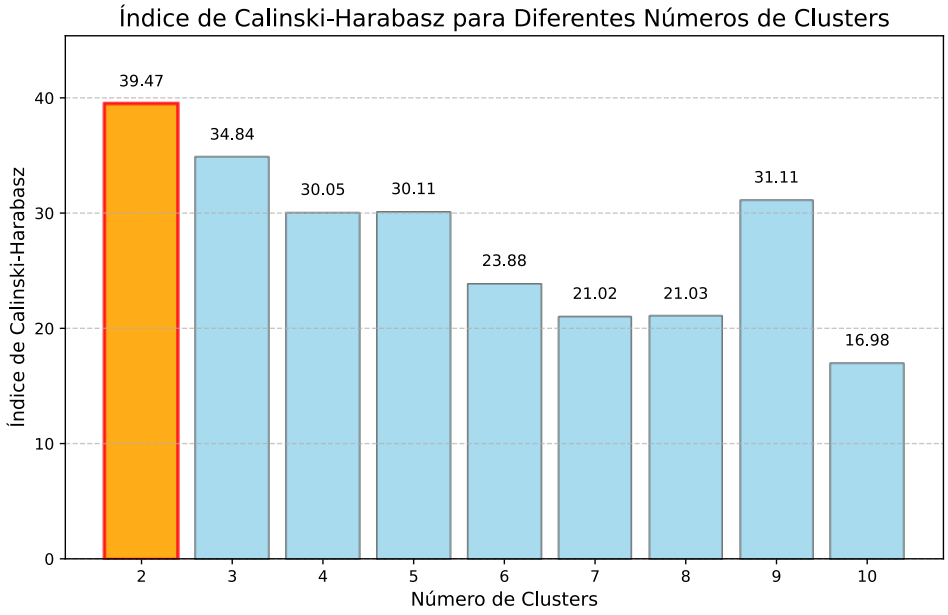


Figura 65. Índice de Calinski-Harabasz para distinto número de *clusters* (k- Medoides).

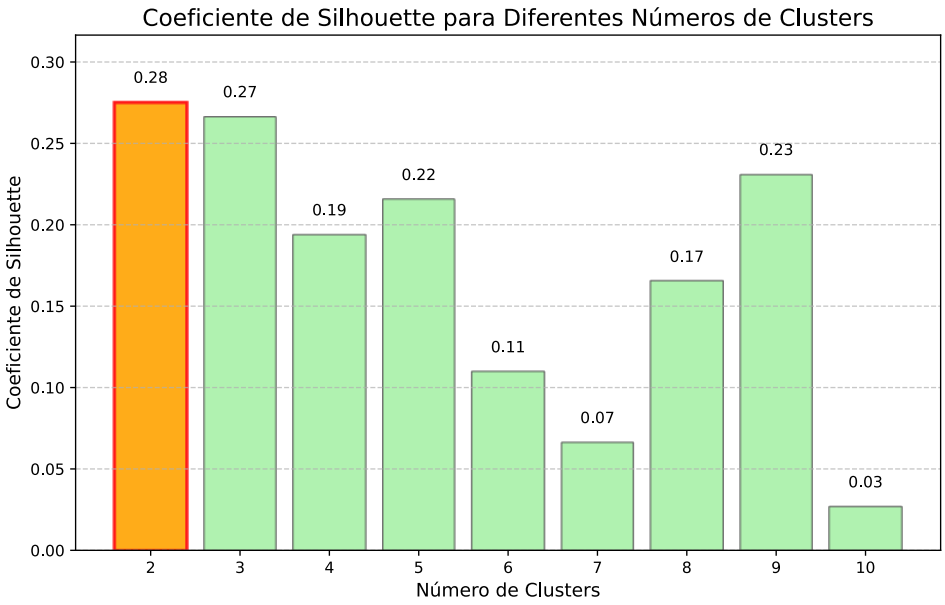


Figura 66. Coeficiente de Silhouette para distinto número de *clusters* (k- Medoides).

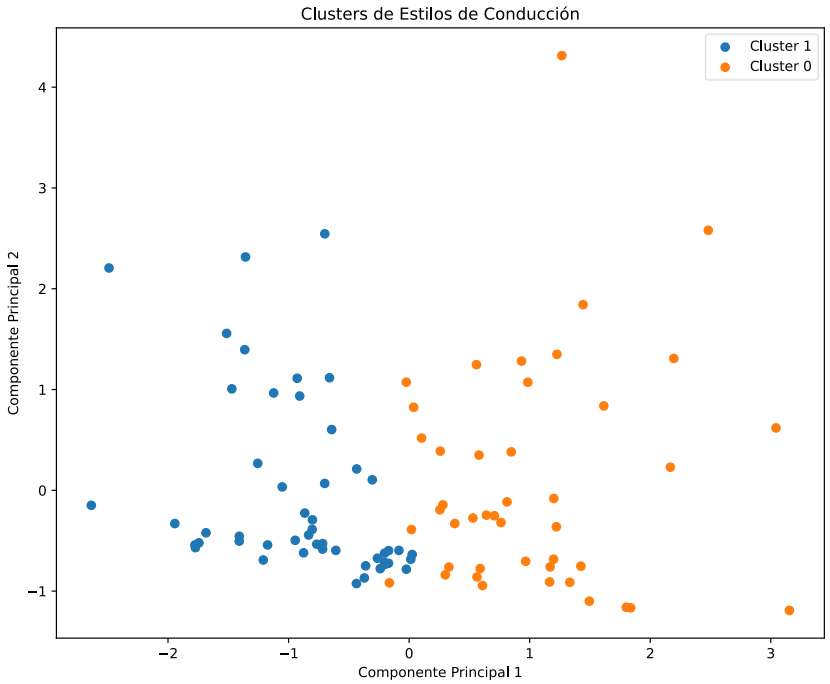


Figura 67. Representación gráfica de los *clusters* (k- Medoides).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.48529 ± 0.12613	0.12 ± 0.13416	0.29642 ± 0.04616
1	0.29672 ± 0.11797	0.06959 ± 0.09998	0.20909 ± 0.04608

Figura 68. Media y desviación típica de las tres *features* en cada uno de los *clusters* (k- Medoides).

ANEXO C. RESULTADOS DESPUÉS DE DATA AUGMENTATION

En este apartado se muestran las gráficas obtenidas con los distintos métodos utilizados para estimar el número óptimo de *clusters* (método del codo, índice de Calinski-Harabasz y coeficiente de Silhouette), así como una representación de los *clusters* tras haber implementado técnicas de *data augmentation*.

C.1. k-Means

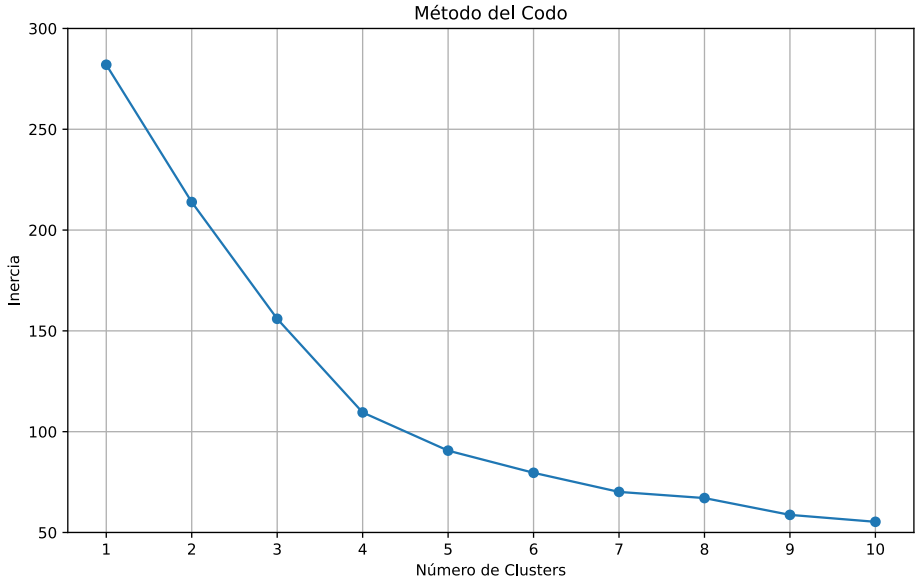


Figura 69. Representación gráfica del método del codo (k-Means).

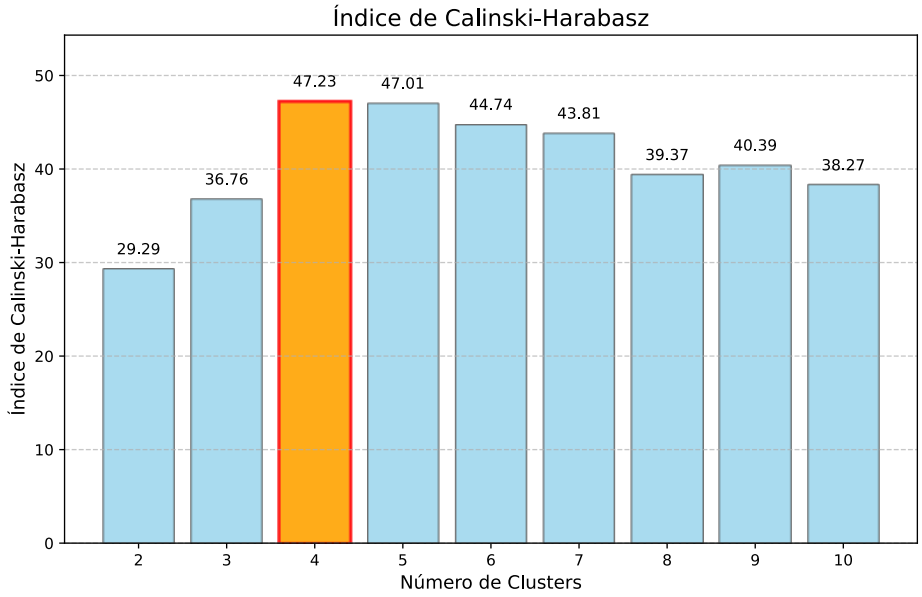


Figura 70. Índice de Calinski-Harabasz para distinto número de *clusters* (k-Means).

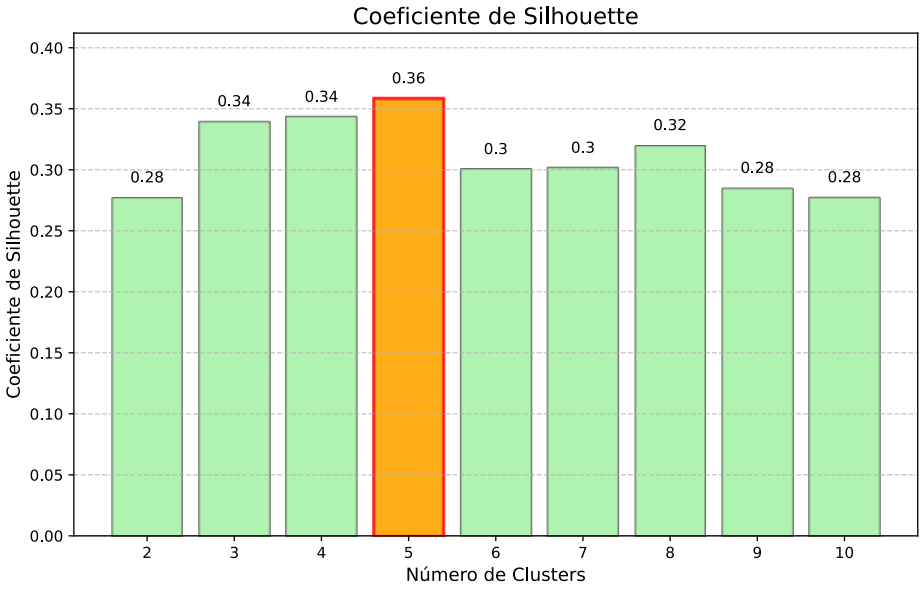


Figura 71. Coeficiente de Silhouette para distinto número de *clusters* (k-Means).

Los métodos utilizados sugieren que podrían existir 4 o 5 *clusters*, por lo que se presentan ambas representaciones para analizar cuál proporciona los mejores resultados.

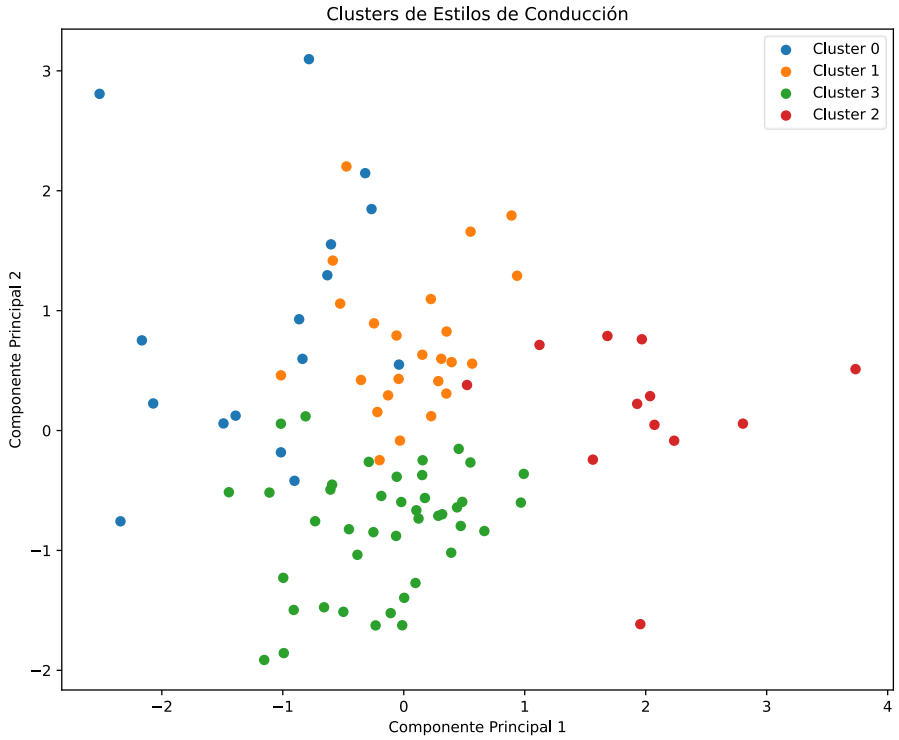


Figura 72. Representación gráfica de los cuatro *clusters* (k-Means).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.37573 ± 0.1633	0.3114 ± 0.11274	0.00457 ± 0.00163
1	0.57039 ± 0.09953	0.07199 ± 0.07162	0.00323 ± 0.0015
2	0.45233 ± 0.1259	0.02577 ± 0.04794	0.01089 ± 0.00269
3	0.28883 ± 0.08202	0.04479 ± 0.05692	0.00415 ± 0.00181

Figura 73. Media y desviación típica de las tres *features* en cada uno de los *clusters* (k-Means).

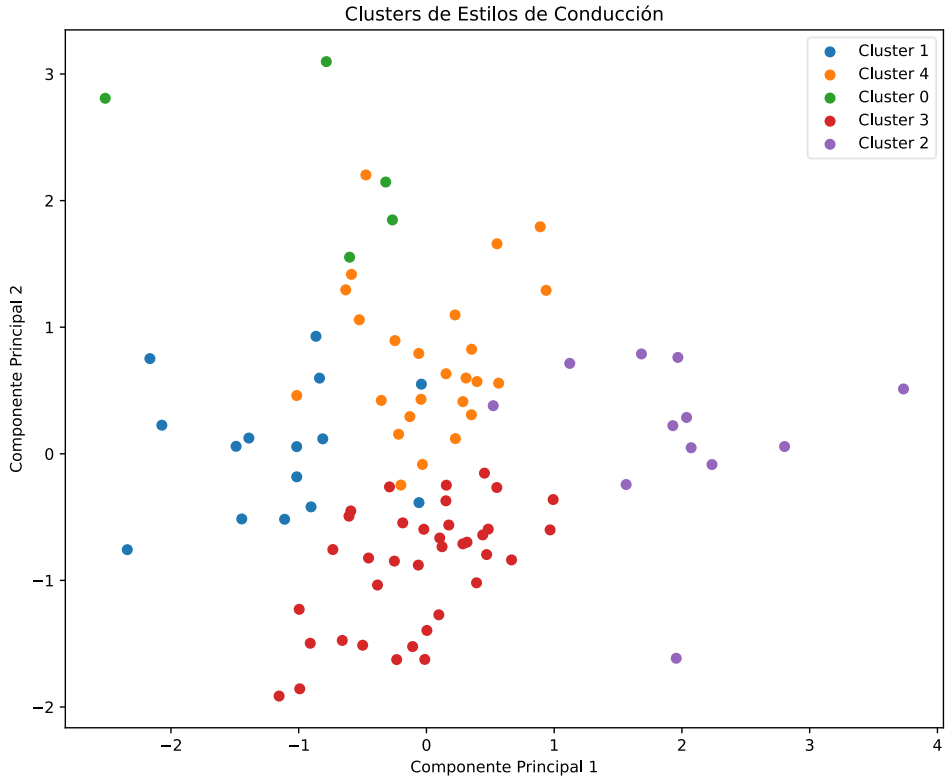


Figura 74. Representación gráfica de los cinco *clusters* (k-Means).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.5478 ± 0.08095	0.40595 ± 0.15268	0.00591 ± 0.00089
1	0.27834 ± 0.0955	0.23804 ± 0.06903	0.00387 ± 0.0018
2	0.45233 ± 0.1259	0.02577 ± 0.04794	0.01089 ± 0.00269
3	0.28925 ± 0.08324	0.02784 ± 0.0343	0.00422 ± 0.00176
4	0.56862 ± 0.09783	0.07844 ± 0.07717	0.00326 ± 0.00147

Figura 75. Media y desviación típica de las tres *features* en cada uno de los *clusters* (k-Means).

C.2. Clustering Espectral

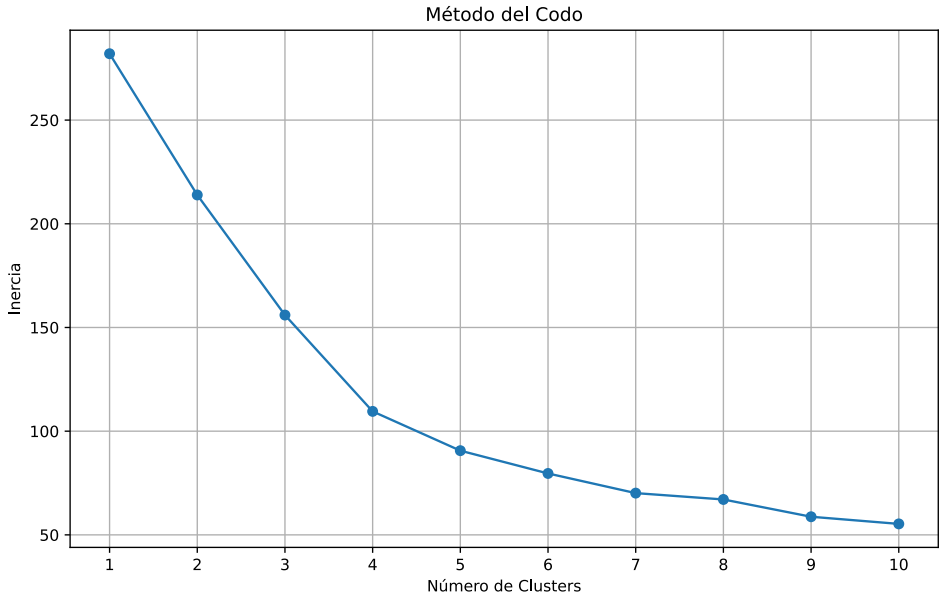


Figura 76. Representación gráfica del método del codo (Clustering Espectral).

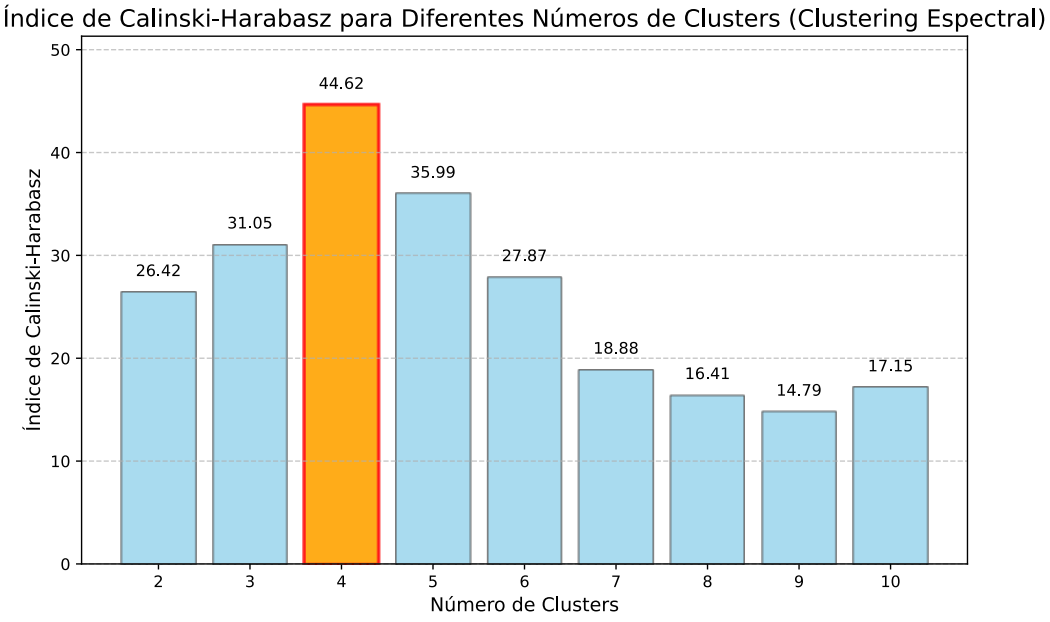


Figura 77. Índice de Calinski-Harabasz para distinto número de *clusters* (Clustering Espectral).

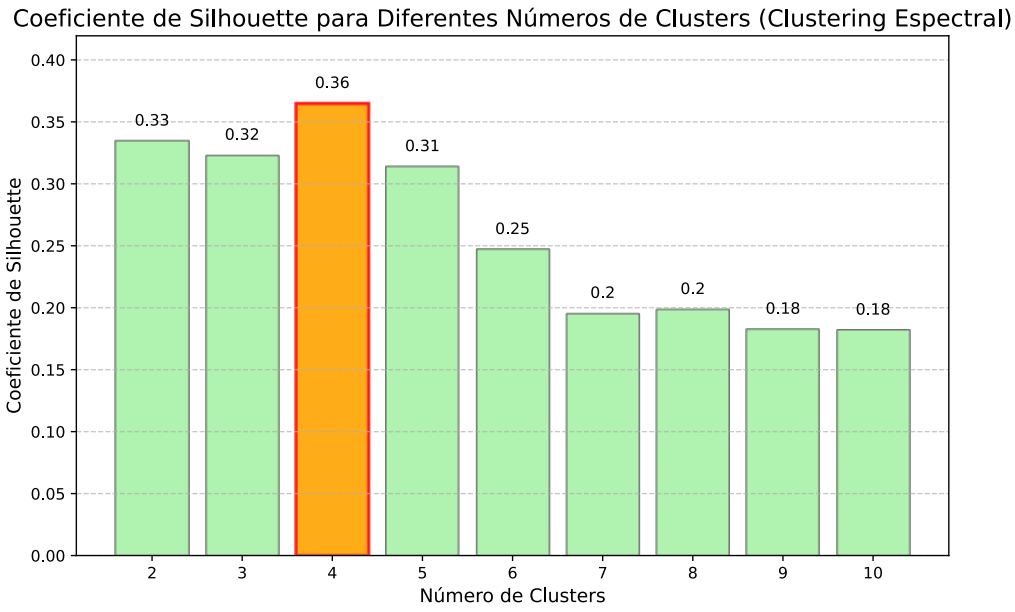


Figura 78. Coeficiente de Silhouette para distinto número de *clusters* (Clustering Espectral).

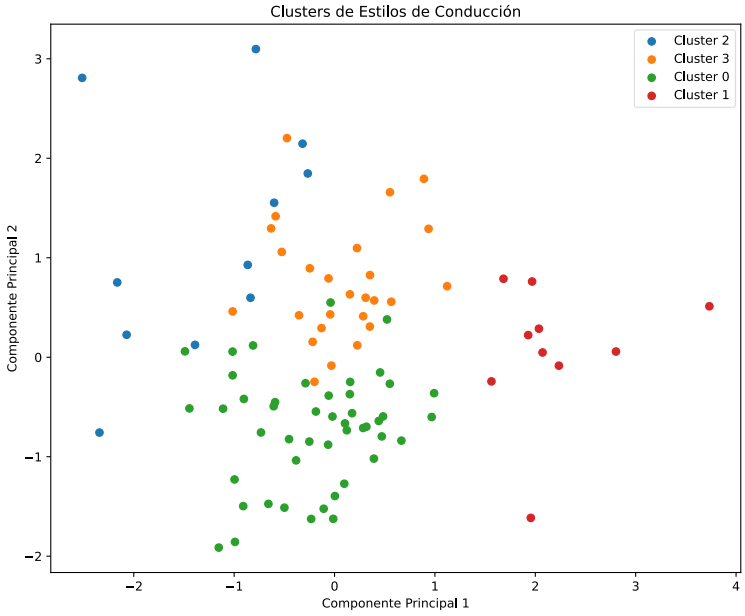


Figura 79. Representación gráfica de los *clusters* (Clustering Espectral).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.29014 ± 0.08122	0.06237 ± 0.07484	0.00429 ± 0.0019
1	0.45002 ± 0.13412	0.00774 ± 0.01218	0.01148 ± 0.00255
2	0.39691 ± 0.17824	0.35108 ± 0.11623	0.00456 ± 0.00159
3	0.56759 ± 0.096	0.07807 ± 0.07564	0.00343 ± 0.00168

Figura 80. Media y desviación típica de las tres *features* en cada uno de los *clusters* (Clustering Espectral).

C.3. Clustering Jerárquico

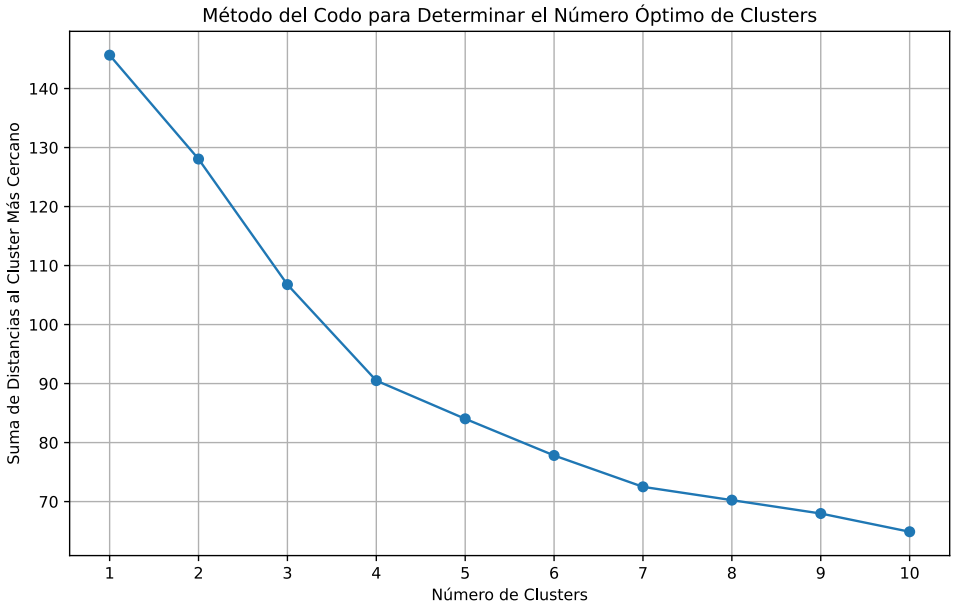


Figura 81. Representación gráfica del método del codo (Clustering Jerárquico).

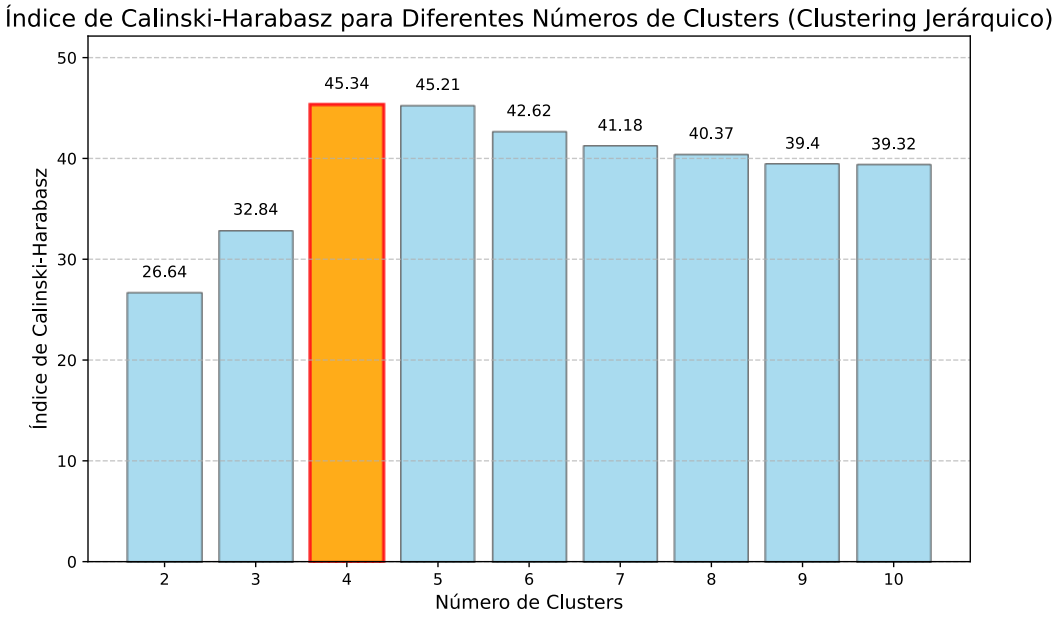


Figura 82. Índice de Calinski-Harabasz para distinto número de *clusters* (Clustering Jerárquico).

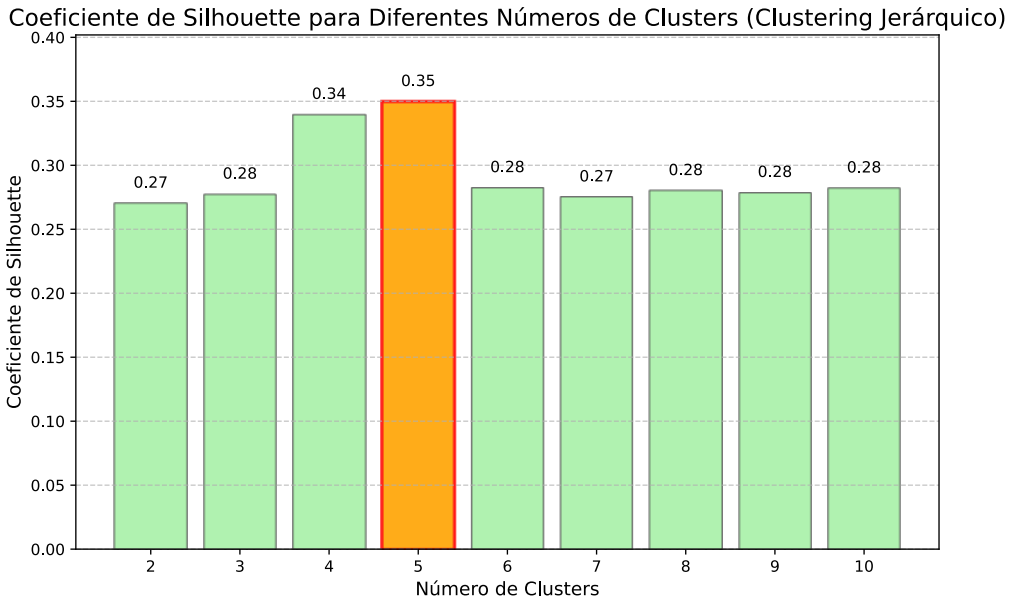


Figura 83. Coeficiente de Silhouette para distinto número de *clusters* (Clustering Jerárquico).

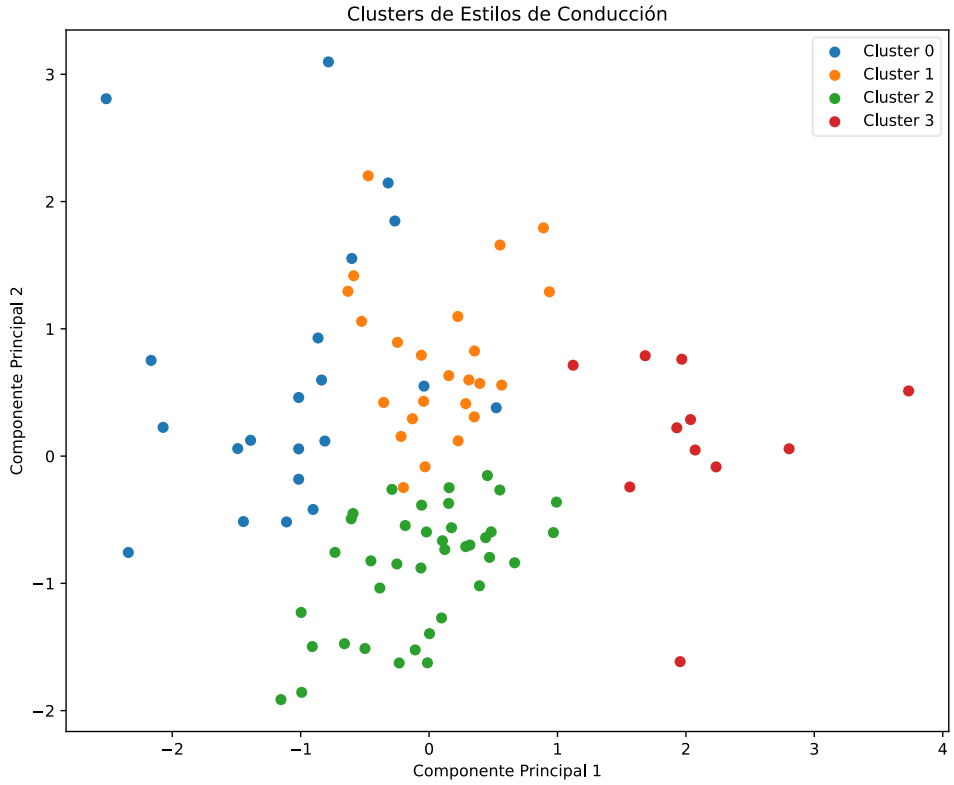


Figura 84. Representación gráfica de los cuatro *clusters* (Clustering Jerárquico).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.35855 ± 0.14509	0.27306 ± 0.11935	0.00427 ± 0.00196
1	0.57343 ± 0.09688	0.07534 ± 0.07723	0.00333 ± 0.00145
2	0.28753 ± 0.08279	0.03189 ± 0.04204	0.00431 ± 0.00182
3	0.45836 ± 0.13021	0.0133 ± 0.02178	0.01113 ± 0.00269

Figura 85. Media y desviación típica de las tres *features* en cada uno de los *clusters* (Clustering Jerárquico).

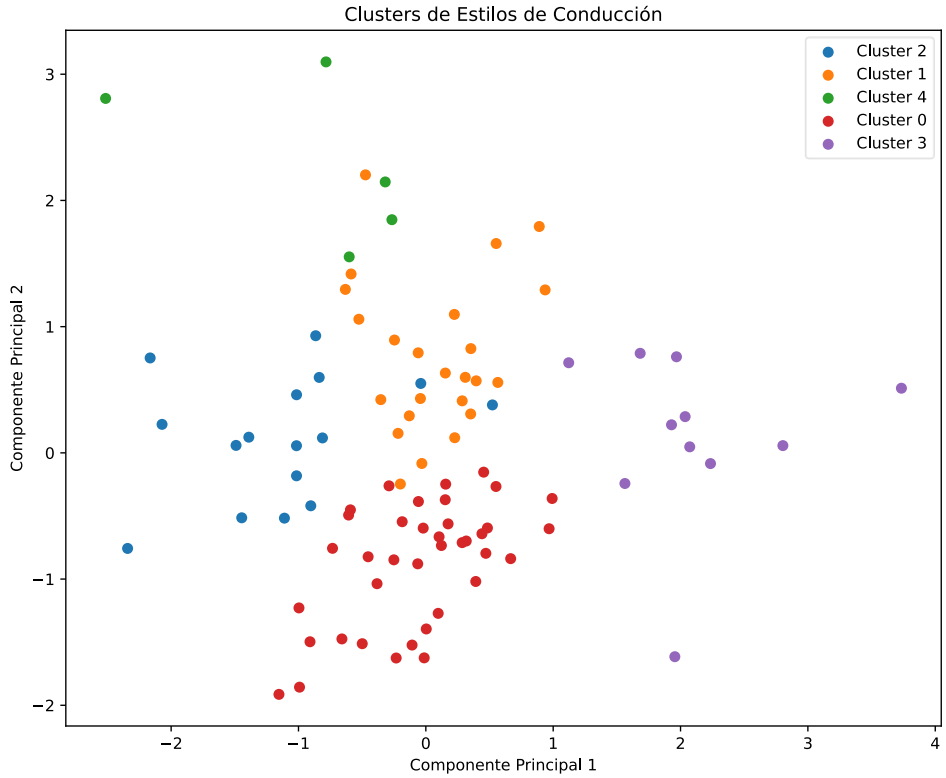


Figura 86. Representación gráfica de los cinco *clusters* (Clustering Jerárquico).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.28753 ± 0.08279	0.03189 ± 0.04204	0.00431 ± 0.00182
1	0.57343 ± 0.09688	0.07534 ± 0.07723	0.00333 ± 0.00145
2	0.29941 ± 0.10321	0.23154 ± 0.07107	0.00376 ± 0.00194
3	0.45836 ± 0.13021	0.0133 ± 0.02178	0.01113 ± 0.00269
4	0.5478 ± 0.08095	0.40595 ± 0.15268	0.00591 ± 0.00089

Figura 87. Media y desviación típica de las tres *features* en cada uno de los *clusters* (Clustering Jerárquico).

C.4. Fuzzy C-Means

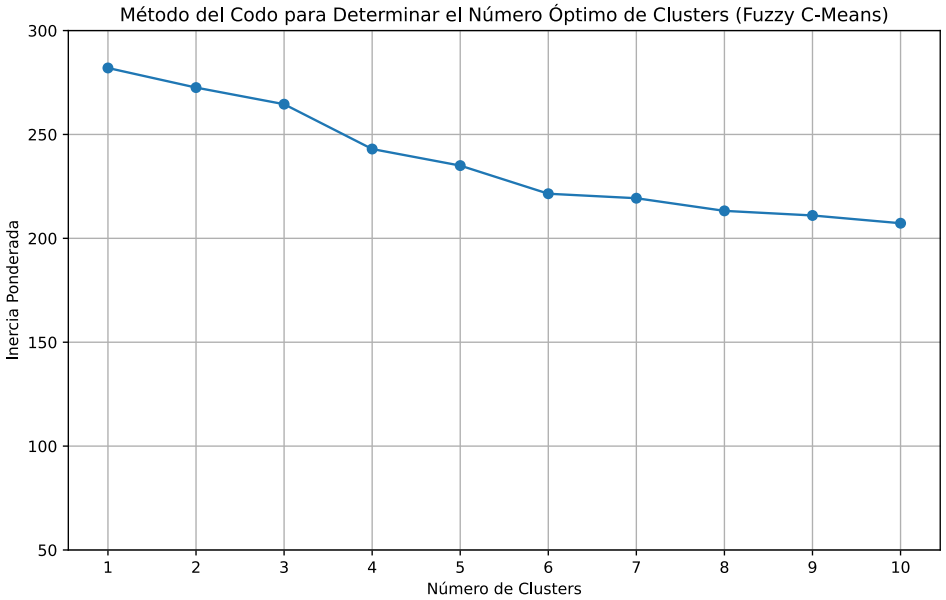


Figura 88. Representación gráfica del método del codo (Fuzzy C-Means).

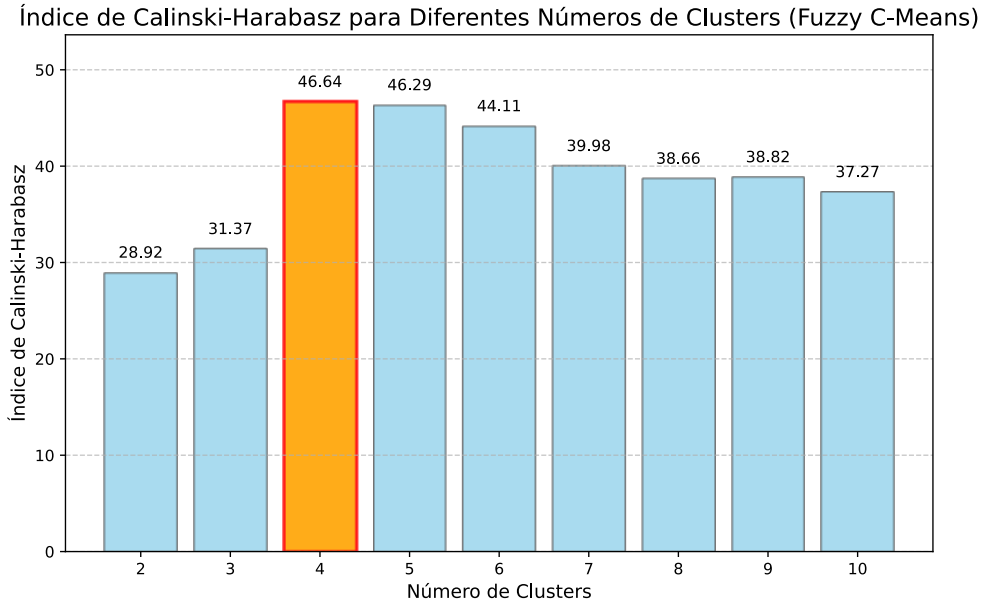


Figura 89. Índice de Calinski-Harabasz para distinto número de *clusters* (Fuzzy C-Means).

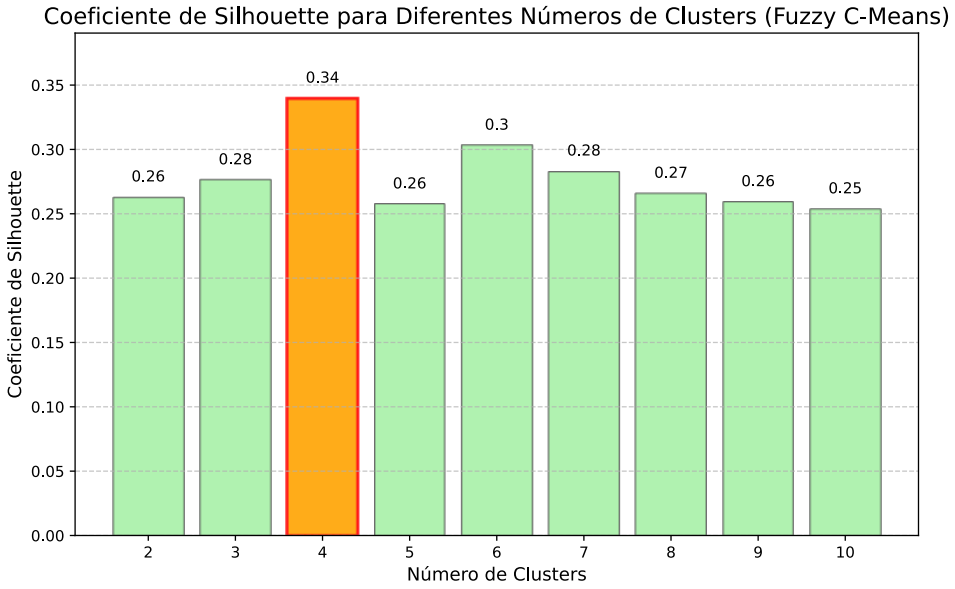


Figura 90. Coeficiente de Silhouette para distinto número de *clusters* (Fuzzy C-Means).

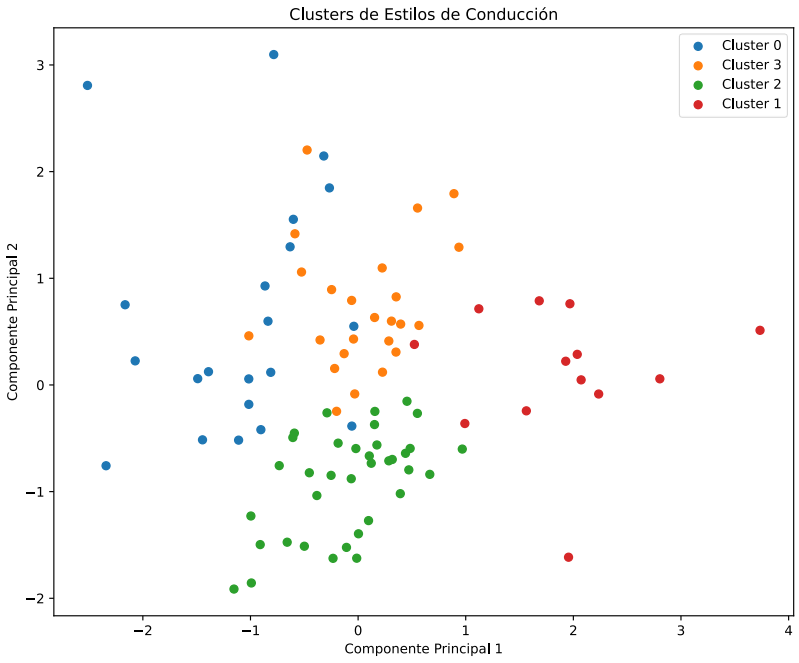


Figura 91. Representación gráfica de los *clusters* (Fuzzy C-Means).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.3543 ± 0.1512	0.27779 ± 0.11571	0.00435 ± 0.0018
1	0.44834 ± 0.12139	0.0238 ± 0.04645	0.01059 ± 0.0028
2	0.28616 ± 0.08224	0.02861 ± 0.03445	0.00414 ± 0.00172
3	0.57039 ± 0.09953	0.07199 ± 0.07162	0.00323 ± 0.0015

Figura 92. Media y desviación típica de las tres *features* en cada uno de los *clusters* (Fuzzy C-Means).

C.5. k-Medoides

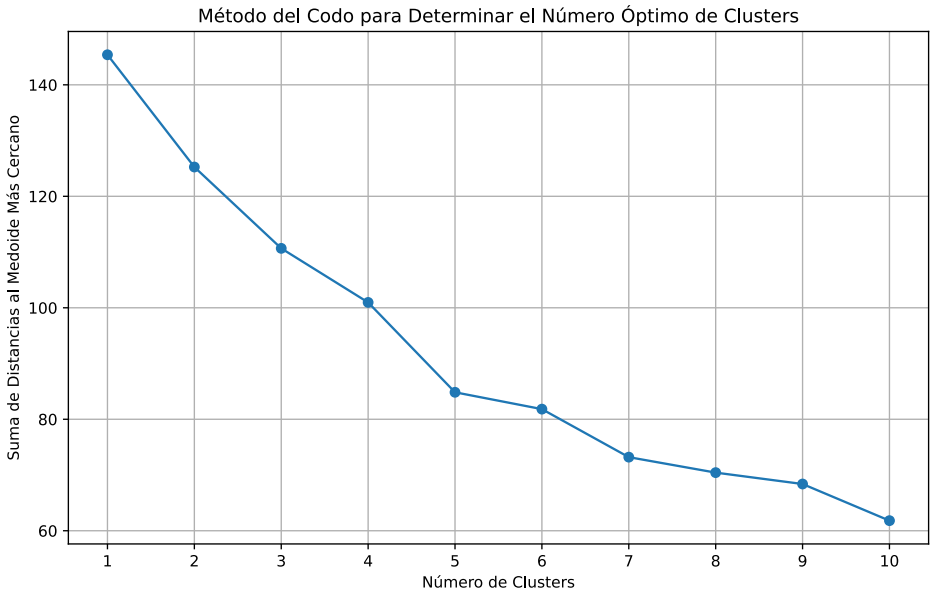


Figura 93. Representación gráfica del método del codo (k-Medoides).

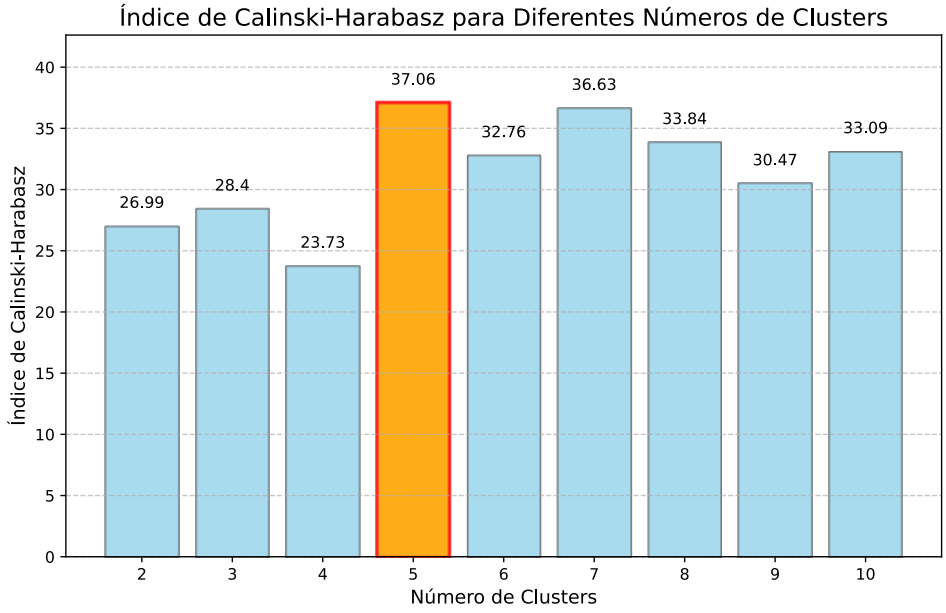


Figura 94. Índice de Calinski-Harabasz para distinto número de *clusters* (k- Medoides).

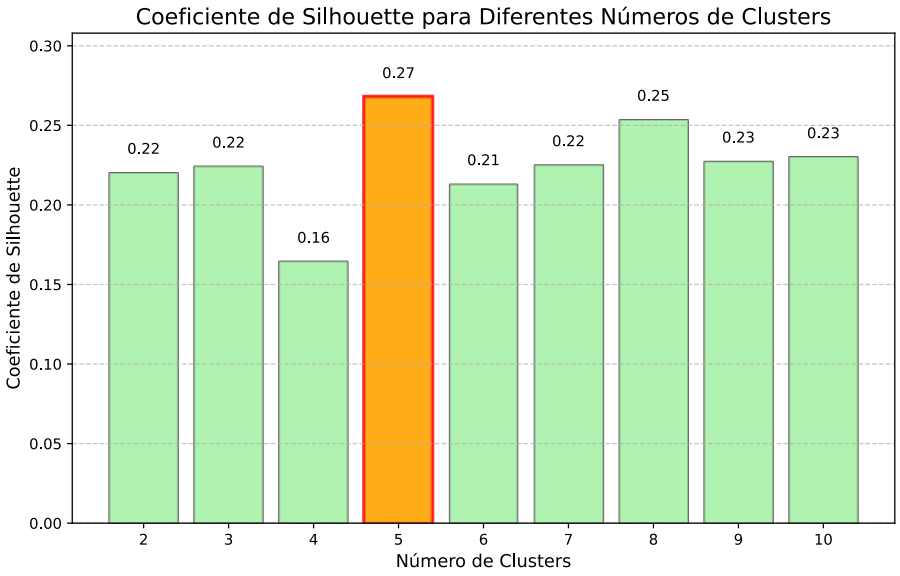


Figura 95. Coeficiente de Silhouette para distinto número de *clusters* (k- Medoides).

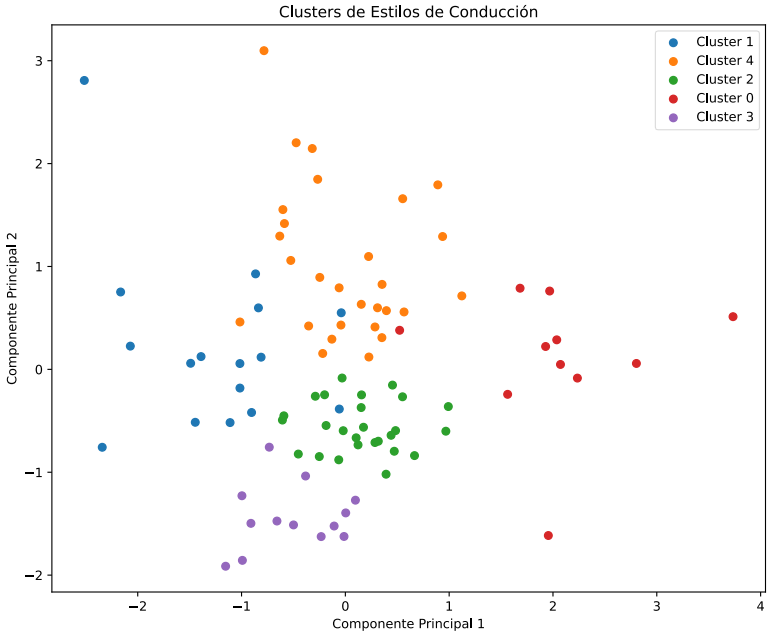


Figura 96. Representación gráfica de los *clusters* (k- Medoides).

cluster	time_over_maxSpeed (mean±std)	phone_usage (mean±std)	hard_acceleration (mean±std)
0	0.4442 ± 0.12869	0.02185 ± 0.04821	0.01119 ± 0.00261
1	0.28779 ± 0.0997	0.26372 ± 0.12247	0.00394 ± 0.00177
2	0.34818 ± 0.05794	0.03082 ± 0.03582	0.00466 ± 0.00166
3	0.19799 ± 0.04728	0.01858 ± 0.02873	0.00303 ± 0.0015
4	0.57653 ± 0.08921	0.12136 ± 0.11942	0.0039 ± 0.00185

Figura 97. Media y desviación típica de las tres *features* en cada uno de los *clusters* (k- Medoides).