

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que me han acompañado durante la realización de este Trabajo de Fin de Grado.

En primer lugar, a la empresa EDICOM, por brindarme la oportunidad de desarrollar este proyecto y facilitarme todos los recursos necesarios. Gracias a esta experiencia, he podido aplicar los conocimientos adquiridos durante la carrera a situaciones reales, enfrentándome a retos que me han permitido crecer tanto personal como profesionalmente.

Mi agradecimiento a Carlos Monserrat, mi tutor en la UPV, por su dedicación, orientación y apoyo constante a lo largo de este trabajo. Su guía ha sido clave para alcanzar la mejor versión posible del proyecto, algo que no habría logrado sin su ayuda.

También quiero agradecer a Carlos Gómez, mi tutor en la empresa, por su apoyo incondicional, sus valiosos consejos y su ayuda fundamental para el desarrollo del proyecto.

Quisiera destacar especialmente a Otman Maamori, mi director de equipo, por su apoyo inquebrantable, su sabiduría y su disposición constante para orientarme y acompañarme en cada etapa de este proceso.

Mi gratitud se extiende a todos los profesores de Ingeniería Informática en la UPV, cuya dedicación y enseñanza me han preparado para afrontar este desafío con confianza y determinación.

Finalmente, gracias a mi familia y a mis amigos más cercanos, por su presencia, ánimo y comprensión en los momentos más exigentes. Su apoyo ha sido fundamental para mantenerme motivado y enfocado hasta el final.

A todos vosotros, muchas gracias.

Resumen

Este proyecto aborda la importancia de comprender los flujos de intercambio electrónico de datos (EDI) para Edicom, una empresa de Software as a Service (SaaS) especializada en integración de datos y automatización de procesos empresariales. Con el crecimiento continuo del sector EDI, también se incrementa la complejidad derivada del volumen de transacciones, la diversidad de clientes y la variabilidad de patrones de uso. En este contexto, a lo largo de este trabajo, se demuestra cómo la analítica avanzada y el aprendizaje automático pueden ser aplicados a los datos de monitorización de transacciones para mejorar significativamente la eficiencia operativa y la toma de decisiones estratégicas.

A lo largo del proyecto, se han desarrollado dos líneas complementarias. La primera consiste en la predicción del volumen de mensajes mediante modelos de deep learning, mientras que la segunda se enfoca en la segmentación de patrones transaccionales usando técnicas de clustering sobre características agregadas. El modelo predictivo basado en redes neuronales recurrentes ha permitido anticipar tendencias de uso y planificar de forma más eficiente los recursos, a pesar de las limitaciones inherentes a los horizontes largos de predicción. Por otro lado, la segmentación mediante K-Means ha logrado identificar perfiles diferenciados de comportamiento entre clientes, abriendo la puerta a una personalización más precisa de los servicios y a una mejor comprensión de los entornos complejos.

Los resultados obtenidos con ambos enfoques se han integrado en la herramienta interna Edicom Analytics, lo que permite visualizar de manera clara los patrones detectados y facilita la extracción de insights accionables. Esta integración refuerza la propuesta de valor de la empresa al incorporar capacidades analíticas y predictivas que optimizan tanto la gestión interna como la interacción con los clientes.

En definitiva, el proyecto no solo ha alcanzado los objetivos planteados, sino que también ha puesto de manifiesto el potencial transformador de la ciencia de datos en contextos reales de negocio. La incorporación de estos modelos en la plataforma de Edicom consolida su posición de liderazgo en el sector y establece una base sólida para futuras mejoras en automatización, personalización y eficiencia de sus servicios.

Palabras clave: deep learning, machine learning, intercambio electrónico de datos, predicciones de carga transaccional, clustering de comportamiento

Abstract

This project addresses the importance of understanding Electronic Data Interchange (EDI) flows for Edicom, a Software as a Service (SaaS) company specialized in data integration and business process automation. With the continuous growth of the EDI sector, the complexity arising from the volume of transactions, the diversity of clients, and the variability of usage patterns also increases. In this context, throughout this work, it is demonstrated how advanced analytics and machine learning can be applied to transaction monitoring data to significantly improve operational efficiency and strategic decision-making.

Two complementary lines have been developed during the project. The first one consists on predicting message volumes using deep learning models, while the second focuses on segmenting transactional patterns using clustering techniques on aggregated features. The predictive model based on recurrent neural networks has allowed anticipating usage trends and more efficiently planning resources, despite the inherent limitations of long-term forecasting horizons. On the other hand, segmentation through K-Means has identified differentiated behavioral profiles among clients, opening the door to more precise service personalization and better understanding of complex environments.

The results obtained with both approaches have been integrated into the internal tool Edicom Analytics, enabling clear visualization of detected patterns and facilitating the extraction of actionable insights. This integration strengthens the company's value proposition by incorporating analytical and predictive capabilities that optimize both internal management and client interactions.

Ultimately, the project has not only achieved the objectives set but also highlighted the transformative potential of data science in real business contexts. The incorporation of these models into Edicom's platform consolidates its leadership position in the sector and establishes a solid foundation for future improvements in automation, personalization, and efficiency of its services.

Key words: deep learning, machine learning, electronic data interchange, transactional load forecasting, behavioral clustering

Índice general

Índice general	9
1 Glosario	11
2 Introducción	13
2.1 Contexto y motivación del proyecto	13
2.2 Objetivos	15
3 Estado del arte	17
3.1 Servicios y componentes clave de Edicom	17
3.2 Otras herramientas de Edicom	17
3.3 Análisis de las herramientas en el mercado	18
4 Análisis del problema	21
4.1 Análisis del marco legal y ético	21
4.2 Análisis de riesgos	22
5 Identificación y análisis de posibles soluciones	23
5.1 Consideración y selección de modelos	24
5.1.1 Modelos de predicción temporal	25
5.1.2 Modelos de clusterización	26
5.2 Propuesta de implementación	27
6 Preparación y análisis de datos	29
6.1 Extracción y creación de la base de datos	29
6.2 Transformación y análisis inicial	30
6.2.1 Limpieza inicial	31
6.2.2 Análisis exploratorio inicial	32
7 Modelos de clustering	47
7.1 Agregación de características y preparación de datos para clustering	47
7.2 Reducción de dimensionalidad	57
7.3 Entrenamiento, evaluación y validación de modelos de segmentación	64
7.3.1 HDBSCAN	65
7.3.2 K-Means	69
7.4 Resultados y selección del modelo final de clustering	74
8 Modelos de predicción	75
8.1 Preparación de datos para el modelo predictivo	75
8.1.1 División de datos para el modelo predictivo	78
8.2 Diseño del modelo	80
8.2.1 Búsqueda de hiperparámetros	84
8.3 Entrenamiento y evaluación del modelo RNN	85
9 Conclusiones y resultados	91
10 Relación del trabajo desarrollado con los estudios cursados y trabajos futuros	93
Bibliografía	95
11 Anexos	103
11.1 Gráficas de interlocutores origen y destino de las transacciones	103

11.2 Resumen de componentes principales	104
11.3 Integración de los datos en las herramientas de Edicom	105
11.4 Edicom Analytics	107
11.4.1 Generación de gráficas de ejemplo con Edicom Analytics	109
11.5 Objetivos de desarrollo sostenible	112

CAPÍTULO 1

Glosario

EDI (Intercambio Electrónico de datos): transferencia estructurada de datos entre organizaciones utilizando formatos estandarizados mediante redes electrónicas.

Mensajes EDI: documentos comerciales y administrativos con un formato específico que debe seguirse para garantizar su correcta interpretación.

Proveedor tecnológico SaaS (Software as a Service): empresa que desarrolla y ofrece software alojado en la nube accesible a través de internet, bajo un modelo de suscripción.

Software de integración: herramienta de conversión de datos internos de una empresa al formato EDI estandarizado y viceversa, asegurando que puedan ser correctamente interpretados por los sistemas receptores.

VANS (Redes de valor añadido): proveedores de servicios de comunicación / almacenamiento que facilitan la transmisión de documentos EDI entre empresas.

Protocolos de comunicación: los protocolos comunes de transmisión en EDI incluyen SFTP, AS2 y métodos de acceso como SOAP y REST, que garantizan la seguridad y autenticidad de los datos.

API (Application Programming Interface): conjunto de reglas que permite que dos aplicaciones se comuniquen entre sí.

Estándares: formatos estándares utilizados en las transacciones definidos por organizaciones internacionales (ODETTE, TRADACOMS, GS1, Peppol y el Comité de Normas Acreditado X12). Algunos son XML, EDIFACT o ANSI X12.

Standard Scaler: Método de normalización de datos que transforma las variables para que tengan una media cero y una desviación estándar uno.

ORDRSP: mensaje de Confirmación de Pedido. Se utiliza para responder a un pedido realizado por el cliente, indicando si se acepta, rechaza o modifica.

INVOIC: mensaje de Factura Electrónica. Representa la solicitud de pago enviada por el proveedor al cliente, detallando los productos o servicios suministrados.

GENRAL: mensaje de Aviso General. Proporciona información general, como notificaciones o actualizaciones relacionadas con transacciones comerciales.

OSTRPT: mensaje de Informe de Estado de Inventario. Permite a las partes interesadas conocer el estado actual del inventario, ayudando en la planificación.

ORDERS: mensaje de Pedido. Se utiliza para realizar una solicitud formal de productos o servicios entre el cliente y el proveedor.

DESADV: mensaje de Aviso de Despacho. Notifica al cliente que se ha enviado un pedido, proporcionando detalles sobre el envío, como el contenido y la fecha de entrega prevista.

RNN (Red Neuronal Recurrente): es un tipo de red de inteligencia artificial que puede recordar información pasada. Se usa para trabajar con datos de serie de tiempo, ya que puede tener en cuenta lo que ocurre en períodos anteriores para mejorar las predicciones futuras.

CAPÍTULO 2

Introducción

El Intercambio Electrónico de Datos se ha consolidado como una herramienta esencial en la era digital, transformando la forma en que las empresas gestionan sus comunicaciones comerciales. Al automatizar el intercambio de documentos electrónicos, como facturas y órdenes de compra, el EDI es capaz de eliminar los procesos manuales y en papel, mejorando la agilidad, precisión y seguridad en las transacciones, mientras reduce el margen de error.

Cada empresa gestiona su flujo de transacciones según sus necesidades específicas, lo que exige flexibilidad y rapidez para adaptarse a cambios comerciales. En este contexto, la extracción de características, el uso de modelos de clustering para identificar patrones transaccionales y el desarrollo de modelos predictivos para variables clave, como el volumen de mensajes intercambiados, son fundamentales para comprender el tránsito particular de cada cliente. Estas herramientas ofrecen una comprensión más profunda de los movimientos de los clientes, lo que contribuye a una gestión y planificación más precisas. Además, proporcionan mayor visibilidad para gestionar de forma eficiente los picos de actividad, al permitir la detección temprana de tendencias, cambios o anomalías en el tráfico. Como resultado, Edicom puede analizar en detalle el comportamiento de los usuarios de sus soluciones, anticiparse a sus necesidades y ofrecer servicios personalizados basados en insights estratégicos.

En definitiva, la adopción de modelos de analítica avanzada de datos establece las bases para una toma de decisiones estratégica que no solo pretende mejorar la eficiencia operativa, sino que también se enfoca en fortalecer la relación con los clientes, brindándoles un servicio más personalizado y proactivo.

2.1 Contexto y motivación del proyecto

La demanda de soluciones EDI modernas, ágiles y escalables está en constante crecimiento, y se espera que el mercado siga expandiéndose. Se proyecta que el mercado de software EDI aumente de 2,81 mil millones de dólares en 2025 a 6,79 mil millones en 2034 [4]. La creciente gestión del volumen de transmisiones, sumada a la interacción con socios globales, la transformación digital en múltiples industrias y la adopción acelerada de transacciones EDI y sus estándares asociados, ha incrementado significativamente la complejidad de los intercambios transaccionales [5], exigiendo enfoques avanzados para su análisis y puesta en marcha, debido a esto, a medida que el volumen de operaciones EDI crece exponencialmente y los patrones de uso varían entre clientes, Edicom enfrenta el desafío de comprender y anticipar estos flujos de trabajo con precisión. Esta complejidad dificulta la gestión eficiente, la identificación de patrones y la recopilación de información estratégica, lo que puede limitar la optimización de recursos, el incremento de ventas, la personalización de servicios y la mejora de la experiencia del cliente.

Para abordar este desafío, Edicom está desarrollando herramientas avanzadas que agilicen el análisis transaccional, como parte de una estrategia más amplia de transformación digital y optimización operativa, impulsada mediante la integración de analítica de datos masivos e inteligencia artificial. En este contexto, la implementación de modelos de segmentación y predicción no solo busca mejorar el análisis de clientes, sino también anticipar la demanda y generar insights estratégicos, facilitando una mayor efectividad

analítica. El acceso a datos sobre tendencias, preferencias y comportamiento del cliente permite a las empresas tomar decisiones más rápidas y fundamentadas, además de diseñar estrategias efectivas para anticiparse a los cambios futuros [26]. Para alcanzar estos objetivos, se han definido dos enfoques complementarios:

- **Predicciones temporales:** la capacidad de prever patrones transaccionales, identificar tendencias y anticipar la evolución del volumen de los intercambios contribuye a una asignación inteligente tanto de recursos tecnológicos como físicos, incluyendo el escalado o la reasignación proactiva de servidores, la personalización de herramientas y el despliegue de nuevas funcionalidades. Asimismo, facilita una planificación más precisa del soporte técnico, la consultoría especializada y las acciones comerciales.

Esta capacidad predictiva mejora significativamente la toma de decisiones estratégicas, al permitir evaluar el impacto de cambios en la plataforma, anticipar el crecimiento de los flujos transaccionales y correlacionar el volumen de transacciones con métricas clave externas como políticas de cobro, satisfacción del cliente o número de incidencias, lo que da lugar a una gestión más informada y proactiva. Todo esto se traduce en una experiencia del cliente más ágil, personalizada y alineada con sus necesidades reales, reforzando la propuesta de valor y la competitividad de la solución.

- **Segmentación basada en patrones transaccionales:** a través de la agregación de características y el análisis de patrones, se busca clasificar las transacciones de los clientes en grupos con comportamientos similares. Esto permitirá comprender mejor las necesidades y particularidades de cada cliente, facilitando el diseño de estrategias de personalización de servicios adaptadas a su actividad [32].

La segmentación de patrones en el tránsito del cliente permitirá a los responsables de gestión de proyectos y a los equipos comercial y de preventa tener una visión analítica más profunda y ágil, reduciendo la necesidad de revisar manualmente los entornos y flujos transaccionales para tener un trasfondo. Esto no solo optimiza el tiempo de análisis, sino que también mejora la eficiencia y proactividad en la toma de decisiones, pudiendo también detectar patrones que a simple vista no se perciben.

La integración de estos modelos en la plataforma de Edicom se propone fortalecer su propuesta de valor, diferenciándose en el mercado al ofrecer capacidades predictivas y analíticas avanzadas. Esto no solo aspira a mejorar la eficiencia operativa interna, sino que también a posicionar a la empresa como un socio estratégico para sus clientes, ayudándolos a optimizar el uso de las herramientas y procesos al tomar decisiones basadas en datos.

En última instancia, este proyecto representa un paso clave en la evolución de Edicom hacia soluciones más inteligentes y adaptativas, consolidando su liderazgo en la integración de datos y la automatización de procesos empresariales. Otros proyectos que han explorado la explotación estratégica de datos empresariales a gran escala para obtener insights estratégicos y mejorar la eficiencia y la toma de decisiones destacan su impacto transformador y los beneficios que este cambio puede aportar [60].

Desde una perspectiva personal, como técnico y consultor, este proyecto representa una valiosa oportunidad para ampliar mis conocimientos y habilidades en áreas fundamentales para nuestras plataformas. Me ha permitido profundizar en el manejo de datos dentro de las herramientas de Edicom, comprender de manera más integral el procesamiento y gestión de los flujos transaccionales e interactuar con tecnologías y sistemas que difieren

de mis entornos y operaciones de trabajo habituales. Además, ha sido una ocasión para reflexionar sobre el impacto directo de las herramientas de tecnologías de analítica avanzada en la toma de decisiones dentro de la gestión de proyectos. Este desafío no sólo ha impulsado mi aprendizaje, sino que también ha sido una fuente constante de motivación, al saber que los resultados pueden generar un valor tangible para la empresa a la vez que para mi crecimiento profesional.

2.2 Objetivos

El propósito de este trabajo es desarrollar un sistema de análisis avanzado que, mediante el uso de técnicas de aprendizaje automático aplicadas a los datos de monitorización, permita anticipar y segmentar los patrones de uso en los flujos transaccionales EDI. Este sistema incorporará modelos predictivos y de segmentación para identificar tendencias tanto en los hábitos operativos como en el volumen de las intercambios EDI de los clientes. El objetivo final es optimizar la toma de decisiones estratégicas, mejorar la asignación de recursos y personalizar los servicios, contribuyendo así a una comprensión ágil, precisa y profunda del entorno empresarial de Edicom, y fortaleciendo la relación comercial con los clientes.

Objetivos:

1. Aplicar técnicas de aprendizaje profundo para predecir el volumen de mensajes EDI con un horizonte temporal amplio, identificando tendencias y fluctuaciones relevantes en los flujos transaccionales.
2. Diseñar e implementar modelos de segmentación mediante clustering para identificar perfiles de comportamiento transaccional entre clientes, basados en variables agregadas y temporales.
3. Integrar los resultados analíticos en herramientas internas (LTA y Edicom Analytics) para facilitar su interpretación visual y permitir la extracción de insights útiles para diferentes áreas de negocio.

CAPÍTULO 3

Estado del arte

En términos generales, una empresa proveedora de servicios EDI se enfoca en convertir y transmitir datos empresariales mediante protocolos de comunicaciones y seguridad, utilizando para ello un software de integración que los adapte a formatos EDI estandarizados, asegurando su correcta interpretación [6].

3.1 Servicios y componentes clave de Edicom

Edicom, como empresa proveedora de servicios SaaS, proporciona una solución integral para la generación, procesamiento y envío o recepción de mensajes EDI de sus clientes. A continuación se detallan sus herramientas y funcionalidades:

- **Ediwin:** es la solución de comunicaciones EDI desarrollada por Edicom, diseñada para gestionar de manera eficiente el intercambio de transacciones comerciales, logísticas y fiscales en grandes empresas [1]. Esta plataforma en la nube, permite a los clientes acceder a la gestión del flujo de mensajes EDI de manera eficiente, ofreciendo una configuración personalizada según sus necesidades.
- **Ebimap:** herramienta para diseñar procesos de transformación de mensajes a distintos formatos, ya sean personalizados o estándar, aplicables tanto a mensajes entrantes como salientes. Esta conversión ocurre a través del proceso de mapeado, que transforma los datos al formato de las aplicaciones de destino y los integra [2].
- **Ebi/ IPaaS:** plataforma en la nube que donde se diseña y configura la transferencia de comunicaciones automatizando las transformaciones necesarias y los módulos para recibir o enviar archivos, a través de ella se integran las aplicaciones como Ediwin y Ebimap junto con los datos que se generan en una empresa [2]. En definitiva, gestiona el envío, recepción, transformación y carga de los mensajes intercambiados por EDI.

Los clientes interactúan exclusivamente con Ediwin, mientras que las otras herramientas son internamente gestionadas por los técnicos que aplican configuraciones personalizadas según los requerimientos acordados, asegurando así agilidad y adaptabilidad.

El éxito de Edicom se basa en su solución integral de gestión de procesos EDI, centralizando los mensajes a través de la plataforma Ediwin, que ofrece una interfaz sencilla y permite una integración flexible y escalable con nuevos socios comerciales. Además, proporciona soporte experto para mantener las operaciones optimizadas.

3.2 Otras herramientas de Edicom

Existen múltiples herramientas desarrolladas por la empresa que, a pesar de no ser obligatorias en un flujo EDI funcional, sí son importantes en el tráfico de integración de este proyecto. Por un lado, se encuentra la plataforma de almacenamiento de forma prolongada **LTA o Long Term Archiving platform**, su uso está enfocado a facilitar a los clientes el almacenamiento y consulta de mensajes o metadatos de una forma prolongada.

Otra de las soluciones clave para este proyecto es **Analytics**. Se trata de una herramienta de análisis y gestión de datos que permite visualizar la información al cliente de forma

clara, accionable y universal. El objetivo es maximizar su estrategia gracias a una mejor monitorización de datos relevantes, [3] que se integran a través de datos almacenados en la plataforma de LTA.

3.3 Análisis de las herramientas en el mercado

Este proyecto propone la integración de métodos de análisis de datos a gran escala para mejorar la comprensión y anticipación de patrones transaccionales en los intercambios EDI, optimizando así la gestión interna y la experiencia del cliente. En el mercado, existen diversas soluciones avanzadas diseñadas para la integración y optimización de transacciones empresariales EDI. Estas soluciones incorporan herramientas de visualización y análisis de datos, así como tecnologías de Machine Learning e Inteligencia Artificial, cada una con enfoques y características específicas que describimos a continuación:

- **Astera:** esta herramienta permite al cliente visualizar de forma accionable los datos EDI sin procesar para extraer información relevante de ellos, mejorando la comprensión de los intercambios de datos [7].
- **IBM Sterling B2B Integrator:** diseñado para integrar y automatizar transacciones B2B mediante EDI, este enfoque prioriza una integración robusta y ofrece análisis avanzados para optimizar operaciones y mejorar la eficiencia en la cadena de suministro. Con IBM Sterling B2B Integration Suite, los datos se presentan de manera accionable, facilitando la comprensión de los intercambios, impulsando el crecimiento empresarial y optimizando la productividad [8], [9].
- **OpenText:** ofrece soluciones que proporcionan supervisión completa en tiempo real [10], [11]. Esta solución propone la inclusión de herramientas de análisis avanzado en la cadena de suministro que permiten a las empresas obtener información valiosa de sus datos, mejorando la toma de decisiones y optimizando procesos en la cadena de suministro.
- **Cleo Integration Cloud:** ofrece herramientas para la integración y visualización de datos en tiempo real, optimizando los procesos de negocio y agilizando la toma de decisiones. Centraliza la recopilación de información en un único lugar y permite personalizar los paneles para que cada usuario acceda solo a los datos relevantes [12], [13].

Las soluciones disponibles en el mercado no solo facilitan el envío y la gestión de transacciones EDI, sino que también permiten a los clientes visualizar datos de su cadena de suministro y obtener insights estratégicos y realizar análisis estratégico a partir de la información EDI. Sin embargo, estas herramientas presentan limitaciones importantes.

Por un lado, algunas soluciones, como OpenText, basan su análisis únicamente en los datos EDI, al depender exclusivamente de esta fuente, caracterizada por su alto nivel de detalle y presencia de información sensible o sesgada, dificulta la identificación de patrones a largo plazo también generando riesgos operativos y de privacidad. Esto afecta la precisión del análisis y posiblemente introduce posibles sesgos en la interpretación, ya que la falta de datos contextuales y adecuados sobre la situación de negocio puede llevar a un ajuste deficiente de los modelos de analítica avanzada y reducir su eficiencia. Es fundamental comprender el significado de los datos y extraer información relevante para garantizar su efectividad [15]. Asimismo, el riesgo no reside en los datos en sí, sino en su interpretación, en las asociaciones que las empresas pueden establecer y en la toma de decisiones automatizada o basada en criterios cuestionables [14].

Por el contrario, otras de las soluciones estudiadas adoptan un enfoque demasiado superficial al limitarse a proporcionar herramientas de visualización sin integrar métodos de procesamiento de datos y analítica a gran escala que aumenten el valor de los datos. Hay que tener en cuenta que Edicom, al igual que otras soluciones en el mercado, ya ofrece herramientas como **Analytics** para visualizar los datos del cliente con el fin de proporcionar insights valiosos. Este proyecto busca reutilizar esa herramienta de visualización analítica para uso interno aportando más valor. De esta manera, se potenciará la efectividad a la hora de transmitir la información de valor obtenida a través de los modelos, asegurando un enfoque más seguro, práctico y estratégico.

Esta propuesta se distingue por centrarse en el análisis de trazas de transferencias EDI de los clientes, una fuente de datos menos compleja y sensible. Esta fuente de datos revela una gran cantidad de información valiosa sobre el funcionamiento del sitio o aplicación, así como el comportamiento del usuario [17], lo que permite obtener insights tanto sobre el funcionamiento interno de las herramientas como sobre el cliente, sin comprometer información crítica del negocio. De esta forma, se agiliza el análisis, se reducen los tiempos de procesamiento y se mejora la comprensión e interpretación de los flujos EDI.

Este análisis busca optimizar las operaciones y mejorar la toma de decisiones dentro de la empresa. La incorporación de modelos de aprendizaje automático no solo permitirá decisiones más informadas y ágiles en el presente, sino que también facilitará la definición de estrategias futuras mediante la identificación de patrones y la estimación de factores como las necesidades de los clientes. En última instancia, esto potenciará el impacto comercial al facilitar un conocimiento más profundo de los clientes, lo que permitirá optimizar y personalizar productos y ofertas [26].

CAPÍTULO 4

Análisis del problema

A medida que aumenta el volumen de operaciones EDI y la diversidad de patrones por cliente, la capacidad de Edicom para comprender y prever adecuadamente los flujos de trabajo disminuye, lo que complica la rapidez de gestión, la identificación de patrones y la toma de decisiones efectivas en tiempo real. La creciente sofisticación y el aumento en el tamaño de las herramientas EDI de Edicom añaden un nivel adicional de complejidad a la gestión y comprensión de estas transacciones. Esta evolución hace que garantizar la robustez y fiabilidad del sistema y sus servicios sea un desafío cada vez mayor, impactando directamente la eficiencia operativa, la toma de decisiones estratégicas y la relación con los clientes en un entorno de intercambios en constante crecimiento [32].

Este vacío genera ineficiencias, ya que la falta de un análisis previo y de hallazgos oportunos sobre las dinámicas de transmisión y las características del cliente dificulta la optimización de recursos. Como resultado, se pierden oportunidades clave para aumentar las ventas, detectar cuellos de botella, mejorar la experiencia del cliente con nuestra herramienta y optimizar los procesos de cobro. Comprender el comportamiento y las necesidades de los usuarios no solo mejora la eficiencia operativa, sino que también permite desarrollar productos y servicios más personalizados, capaces de satisfacer mejor sus expectativas e incluso revelar necesidades que aún no se habían identificado [26].

4.1 Análisis del marco legal y ético

El cumplimiento normativo, especialmente en lo relativo al Reglamento General de Protección de Datos (GDPR) y a las directrices de organismos nacionales como la Agencia Española de Protección de Datos (AEPD), es esencial para garantizar la protección de los derechos de las personas y entidades cuyos datos son procesados. Este apartado analiza los principios legales y éticos que rigen este proyecto, resaltando las medidas implementadas para asegurar su cumplimiento.

En este proyecto, los datos procesados provienen exclusivamente del monitoreo del flujo de mensajes EDI y, en general, no incluyen información personal identifiable. Sin embargo, existen casos en los que el NIF de los clientes puede quedar expuesto. Aunque la cantidad de datos sensibles es limitada, es fundamental garantizar la protección de los derechos y libertades de los interesados mediante procesos de anonimización como en los de seudonimización [18]. El proceso de anonimización busca eliminar o reducir al mínimo el riesgo de reidentificación de los datos, asegurando al mismo tiempo la precisión y fiabilidad de los resultados obtenidos en su tratamiento [19].

Dado que los datos han sido completamente anonimizados y el proyecto tiene, por el momento, un enfoque exclusivamente investigador, sin impacto directo en la toma de decisiones, no es necesario obtener el consentimiento explícito de los usuarios. No obstante, se han adoptado medidas estrictas para garantizar que los datos permanezcan anonimizados y evitar cualquier posibilidad de reidentificación. Asimismo, se han implementado protocolos de seguridad para proteger su confidencialidad, incluyendo controles de acceso y políticas rigurosas de gestión de datos, con el fin de prevenir accesos no autorizados o posibles brechas de seguridad. Es fundamental adoptar las precauciones necesarias para mitigar los riesgos de reidentificación y divulgación de la información [20].

En relación con los modelos predictivos, se ha tomado especial cuidado para evitar cualquier tipo de discriminación, tanto directa como indirecta, hacia cualquier grupo. Se ha evaluado que no existan atributos sensibles que puedan influir de manera inapropiada en las decisiones automatizadas, especialmente en áreas como la fijación de precios o el acceso a servicios.

4.2 Análisis de riesgos

El análisis de riesgos es fundamental para identificar, evaluar y mitigar posibles problemas que puedan comprometer la implementación y operación del proyecto. La obtención de resultados incorrectos puede impactar negativamente en las estrategias de la organización y la experiencia del cliente, generando ineficiencias y sobrecostos. Es crucial mantener un Retorno de Inversión (ROI) positivo. Esta métrica mide la rentabilidad de un proyecto, calculando el beneficio económico generado en relación con la inversión realizada [21].

Para garantizar resultados óptimos, es fundamental evaluar y validar continuamente el modelo predictivo, permitiendo así su mejora constante. Dado que los modelos predictivos no son estáticos, es necesario monitorearlos y ajustarlos de manera periódica para mantener su precisión y eficacia [23], [25], [22]. Esto incluye realizar pruebas de calidad, emplear modelos interpretables siempre que sea posible, comunicar claramente las limitaciones del sistema y reentrenar el modelo periódicamente. También es necesario considerar los riesgos asociados a la calidad e inconsistencia de los datos y su impacto en la duración y los resultados del proyecto. Problemas como lagunas, inconsistencias y duplicidades pueden comprometer el éxito de un proyecto de análisis de datos e incluso llevarlo al fracaso [24].

Finalmente, la tasa de adopción de la solución, su nivel de uso y la precisión de su análisis son factores clave para determinar el éxito del proyecto. Por ello, es importante evaluar el riesgo de que estos aspectos no se cumplan [21]. Para que ésta aumente, es fundamental asegurar que el modelo pueda adaptarse a un mayor volumen de datos, realizar análisis de impacto, establecer planes de contingencia y definir métricas de satisfacción que permitan evaluar su aceptación y efectividad.

CAPÍTULO 5

Identificación y análisis de posibles soluciones

Este trabajo busca abordar el desafío que supone la creciente complejidad y el alto volumen de transacciones EDI de los clientes, los cuales dificultan un análisis eficiente de los flujos transaccionales. Esta situación limita la capacidad de Edicom para tomar decisiones informadas que optimicen recursos, identifiquen tendencias y minimicen inefficiencias operativas. No obstante, el uso de análisis avanzado de datos puede mejorar significativamente la toma de decisiones, reducir riesgos y revelar patrones valiosos que, de otro modo, pasarían desapercibidos [26].

Para alcanzar los objetivos de este proyecto, es fundamental tanto para la segmentación como para la predicción, seleccionar modelos de análisis de datos adecuados basando la elección en criterios sólidos y bien fundamentados. La combinación de técnicas de segmentación y análisis predictivo no solo permite una mejor comprensión de los datos históricos, sino que también facilita la toma de decisiones proactivas, ayudando a reducir riesgos y mejorar la eficiencia operativa [29], estas herramientas están revolucionando la toma de decisiones empresariales, permitiendo a las organizaciones comprender mejor a sus clientes, anticipar cambios en el mercado y optimizar sus operaciones de manera más eficiente [28]. Asimismo, el análisis masivo de datos históricos facilita la identificación de patrones ocultos, mejorando la precisión tanto en la segmentación como en las predicciones [30].

La integración de técnicas avanzadas de minería de datos ha demostrado ser esencial en otros sectores en los que se trabajan con transmisiones como el de las telecomunicaciones, para comprender dinámicas de negocio, identificar patrones y mejorar la competitividad en el sector [32].

El éxito de estos modelos depende, en gran medida, de una preparación rigurosa de los datos, que incluye su agregación y la identificación de características clave para reconocer patrones significativos. Para enriquecer los datos, mejorar su variabilidad y aumentar su capacidad de segmentación, es fundamental considerar las agrupaciones basadas en las variables categóricas presentes en el conjunto inicial:

- **Tipo de mensaje:** facilita el análisis de patrones transaccionales entre clientes.
- **Dirección del mensaje:** ayuda a entender el flujo de intercambios.
- **Tuplas de Interlocutores:** permite analizar interacciones específicas del cliente.
- **Módulo de seguridad/ envío:** identifica patrones de configuración y perfiles transaccionales.
- **Categoría de tamaño del mensaje:** detecta anomalías y tendencias en el tamaño.

A partir de estas agrupaciones y las variables originales, se pueden crear características adicionales o transformar las ya existentes para permitir capturar mejor los patrones subyacentes de los datos [42], estas pueden ser el tiempo transcurrido desde el último

mensaje (**Time Since Last Message - TSLM**), el número de mensajes, y métricas estadísticas como la media, la desviación estándar y el **Z-Score**. También es posible calcular ratios, tasas transaccionales y tiempos de espera entre mensajes.

En cuanto a la periodicidad, resulta conveniente agrupar los datos de la serie temporal por períodos, como hora, día, semana o mes. Aunque esta agregación reduce el nivel de detalle, permite regularizar la periodicidad de serie temporal, minimizando el ruido al suavizar fluctuaciones aleatorias que podrían dificultar su interpretación. Además, mejora la manejabilidad de los datos al reducir su volumen y complejidad, lo que facilita la identificación de tendencias y patrones subyacentes [32].

Debido a la naturaleza temporal de los datos, para capturar relaciones temporales, se pueden incluir variables cíclicas que representen la periodicidad, así como incorporar rezagos temporales basados en patrones de correlación significativos [31]. Los movimientos cíclicos y las variaciones estacionales son características clave en las series temporales y resultan fundamentales para el análisis y modelado predictivo [32]. Igualmente, la ingeniería de características basada en el tiempo permite extraer información valiosa a partir de la fecha y la hora de los datos. Variables como el día de la semana, la hora del día o el mes del año pueden reflejar la estacionalidad y los patrones temporales de la serie, mejorando así la interpretación de los datos y la precisión de los modelos predictivos [31].

Asimismo, el uso de variables de rezago es una técnica ampliamente utilizada en el análisis de series temporales, ya que permite capturar dependencias temporales y tendencias en los datos. Incorporar valores previos de la serie temporal como características puede mejorar la capacidad del modelo para identificar patrones de estacionalidad y tendencias a largo plazo. También se han considerado que las estadísticas de ventana móvil ayudan a suavizar el ruido y resaltar las tendencias subyacentes en los datos, lo que facilita la detección de cambios y anomalías [31].

Un preprocesamiento adecuado es fundamental para alinear los datos con los objetivos del análisis y las características del modelo, garantizando resultados óptimos. El principal desafío es lograr un equilibrio entre conservar los detalles relevantes y estructurar los datos de manera que sean manejables y funcionales para el modelo. Sin embargo, la integración de las técnicas mencionadas permitirá mejorar la calidad del análisis y maximizar la precisión de las predicciones, facilitando la toma de decisiones basada en datos [31].

5.1 Consideración y selección de modelos

En este proyecto, la selección adecuada de modelos es crucial para abordar los desafíos derivados de la creciente complejidad y el volumen de las transacciones EDI. Estos datos requieren enfoques que no solo sean capaces de procesar grandes volúmenes de información, sino que también permitan extraer insights significativos para una toma de decisiones más informada. El análisis de datos en tiempo real juega un papel clave en la identificación de patrones estratégicos en el comportamiento de los clientes y del mercado, optimizando así la toma de decisiones y la planificación empresarial [26].

La selección de modelos debe basarse en la naturaleza de los datos y en los objetivos específicos del proyecto. En este caso, al tratarse de logs de transacciones registradas a lo largo del tiempo, los datos tienen la estructura de una serie temporal. Por otro lado, además de la precisión, es fundamental considerar cuatro criterios clave como lo son la **velocidad**, que mide los costos computacionales de generar y utilizar el modelo; **robustez**, que evalúa su capacidad para hacer predicciones correctas ante datos ruidosos o incompletos; **escalabilidad**, que analiza su eficiencia al procesar grandes volúmenes de

datos; e **interpretabilidad**, que determina la facilidad de comprensión y extracción de conocimiento del modelo [32].

Inicialmente, se ha adoptado un enfoque de modelado predictivo, considerando el intercambio continuo de observaciones en los datos de monitorización de transacciones EDI. El foco se ha puesto en modelos diseñados para capturar la naturaleza temporal dominante de estos datos y la posible relación a largo plazo entre las observaciones. Modelar con precisión la evolución futura del número de transacciones es fundamental para que Edicom pueda anticiparse a situaciones como picos de demanda o posibles cuellos de botella. La predicción mediante series temporales permite identificar tendencias a largo plazo, variaciones estacionales y patrones cíclicos en los datos, lo que facilita una planificación eficiente de recursos y la detección temprana de anomalías en los flujos transaccionales [32]. Esto permite adoptar estrategias proactivas para mantener la eficiencia operativa y optimizar la asignación de recursos. Además, anticipar fluctuaciones en el volumen de transacciones y estimar la demanda futura es fundamental para la planificación estratégica, lo que resulta clave para el éxito del negocio [28].

Se ha aplicado además, el clustering en series temporales para segmentar y detectar patrones o comportamientos similares en distintos grupos de transacciones de los clientes. Este enfoque permite identificar dinámicas subyacentes que no son inmediatamente evidentes, como períodos de mayor actividad o tipos de transacciones propensas a generar ineficiencias.

Agrupar transacciones con patrones horarios similares no solo optimiza la gestión de recursos, sino que también facilita la personalización de servicios y mejora la eficiencia operativa. Además, el clustering desempeña un papel clave en la segmentación de datos y en la inteligencia de negocios, ya que organiza grandes volúmenes de información en grupos significativos, favoreciendo así la toma de decisiones estratégicas y fortaleciendo la relación con los clientes [32], ya que la segmentación de poblaciones permite ofrecer servicios diferenciados, optimizando la distribución de recursos y la personalización de soluciones [27].

5.1.1. Modelos de predicción temporal

Se han evaluado diversos enfoques, considerando modelos clásicos de series temporales como **ARIMA**, **SARIMA** y **VAR**. Sin embargo, estos fueron descartados debido a sus limitaciones ya que, aunque son eficaces para capturar relaciones lineales y patrones estacionales, estos asumen que el valor futuro de una variable depende linealmente de observaciones pasadas junto con errores aleatorios, lo que limita su capacidad para capturar la variabilidad inherente a ciertos datos. Además, su precisión en predicciones a largo plazo puede verse afectada, especialmente ante cambios estructurales o estacionalidades poco definidas [61].

Dado el contexto de los datos, se ha enfocado el proyecto en los modelos de **deep learning**, como **LSTM (Long Short-Term Memory)**/ **GRU (Gated Recurrent Unit)**. Estos modelos son capaces de capturar dependencias temporales profundas, así como relaciones no lineales y dinámicas, sin requerir una extensa ingeniería manual de características, ya que aprenden directamente de las secuencias temporales. Gracias a su arquitectura recurrente, son especialmente eficaces para identificar y modelar relaciones a largo plazo dentro de los datos, gestionando de manera eficiente patrones temporales complejos. Esta capacidad los convierte en herramientas particularmente útiles para realizar predicciones precisas incluso en escenarios con alta complejidad [33].

Además de ofrecer una mejor gestión de la variabilidad y la no linealidad en las series temporales. Su escalabilidad los hace adecuados para predicciones a largo plazo y hasta de múltiples variables. A diferencia de los modelos tradicionales, cuya precisión puede

degradarse al aumentar el volumen de datos, las redes neuronales profundas tienden a mejorar su rendimiento con conjuntos de datos más amplios, ya que esto les permite aprender patrones más detallados y complejos con mayor precisión [46].

No obstante, estos modelos también presentan ciertas limitaciones, como la necesidad de contar con grandes volúmenes de datos para lograr un entrenamiento adecuado y el riesgo de sobreajuste [51]. A pesar de estos desafíos, su capacidad para procesar datos secuenciales de forma eficiente ha consolidado su adopción en tareas de predicción de series temporales.

En particular las **LSTM** incorporan celdas de memoria y mecanismos de compuerta especializados que les permiten retener información relevante durante períodos prolongados, facilitando el modelado de relaciones dinámicas y patrones irregulares en las series temporales [34]. Estos mecanismos permiten controlar el flujo de información dentro de la red, lo que mejora la capacidad de las **LSTM** para capturar relaciones de largo plazo [33].

Las redes **GRU**, por otro lado, surgen como una alternativa simplificada a las **LSTM**, reduciendo su complejidad computacional sin sacrificar significativamente el rendimiento. A diferencia de las **LSTM**, las **GRU** combinan las funciones de la puerta de entrada y la puerta de olvido en un solo mecanismo, lo que disminuye el número de parámetros y acelera el entrenamiento del modelo [35]. Esto las hace especialmente útiles cuando se trabaja con volúmenes de datos limitados o cuando se requiere un menor costo computacional [36], [37].

5.1.2. Modelos de clusterización

La clusterización es el proceso de particionar un conjunto de datos en subconjuntos denominados clusters, de manera que los objetos dentro de un mismo cluster sean similares entre sí y disimilares a los objetos en otros clusters [32].

Se han estudiado diversos modelos de clustering, como **GMM**, **Spectral**, **Agglomerative Clustering**, **OPTICS** y **DBSCAN**. Sin embargo, finalmente este proyecto se centra en el modelado de **K-Means** y **HDBSCAN** ya que ofrecen un equilibrio entre eficiencia, escalabilidad y flexibilidad. **K-Means** es un método eficiente, ideal para procesar grandes volúmenes de datos con un costo computacional moderado. Por otro lado, **HDBSCAN** es más flexible, ya que permite identificar clusters de formas irregulares y con densidades variables, sin depender exclusivamente de la distancia euclídea, lo que permite adaptarse a la estructura de los datos y segmentarlos de manera efectiva, incluso en escenarios con alta variabilidad [32]

- **K-Means:**

Es un método de clustering particional que agrupa los datos en un número predefinido de clusters, asignando cada punto al cluster cuyo centroide sea más cercano. Es un algoritmo eficiente, con una complejidad temporal de $O(nkt)$, donde n es el número de objetos, k el número de clusters y t el número de iteraciones [32].

Este modelo es adecuado para grandes volúmenes de datos, especialmente cuando los clusters son de forma esférica y tienen densidades similares. Sin embargo, presenta varias limitaciones, ya que requiere que el número de clusters se defina previamente, es sensible a la inicialización de centroides y no es ideal para clusters con formas arbitrarias o densidades variables [53].

- **HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise):**

Es un método basado en densidad que extiende **DBSCAN** al incorporar una jerarquía de clusters y un criterio de estabilidad para determinar los conglomerados

[62]. A diferencia de los métodos particionales como **K-Means**, no requiere especificar el número de clusters de antemano, sino que identifica automáticamente regiones de alta densidad en el espacio de datos y descarta puntos que considera ruido.

HDBSCAN construye un árbol de expansión mínimo del conjunto de datos, lo que le permite formar una jerarquía de clusters y podarla para identificar los grupos más estables [63]. Esto le permite detectar conglomerados de diferentes escalas y manejar datos ruidosos de manera eficiente. Además, es particularmente útil en datos de forma arbitraria y densidad variable.

Sin embargo, este modelo presenta algunos desafíos. La selección de hiperparámetros como *min_cluster_size* y *min_samples* puede ser compleja, y su rendimiento puede verse afectado en conjuntos de datos de alta dimensionalidad. Además, a diferencia de **K-Means**, no garantiza que todos los puntos sean asignados a un cluster, ya que algunos pueden ser clasificados como ruido [63].

En resumen, la selección del modelo de clustering adecuado depende del tipo de datos y los objetivos del análisis. Mientras que **K-Means** es eficiente y fácil de interpretar, **HDBSCAN** ofrece mayor flexibilidad al adaptarse a estructuras de clusters complejas y permitir la detección de ruido.

5.2 Propuesta de implementación

Siguiendo los criterios definidos en la sección anterior, se han seleccionado los modelos a implementar. En primer lugar, se optará por redes neuronales **LSTM** y **GRU**, debido a su alta precisión y capacidad para manejar grandes volúmenes de datos con dependencias temporales a pesar de su coste computacional. Por otro lado, para la tarea de segmentación de grupos, se evaluarán los modelos **K-Means** y **HDBSCAN**, cuya efectividad se medirá en función de la separación obtenida, así como su precisión y fiabilidad.

Para la preparación de los datos con los que alimentar a los modelos, es necesario agrupar todas las observaciones en intervalos horarios para trabajar con un volumen regular y significativo de períodos, lo que permite detectar variaciones rápidas y cambios en la tendencia o estacionalidad en el flujo de las transacciones [32]. Asimismo, se busca agregar nuevas variables en función de los objetivos específicos de cada modelo. El conjunto de datos diseñado para el modelo predictivo, estará centrado en el número de mensajes transaccionados y sus dependencias temporales. Por su parte, el modelo de clusterización incorporará características diferenciadoras que permitan una segmentación efectiva de las clases en distintos perfiles transaccionales.

En ambos enfoques se busca alcanzar la mayor precisión posible. Para lograrlo, será necesario definir, evaluar y validar tanto los hiperparámetros como la estructura de los modelos, con el fin de identificar la combinación que mejor se adapte a las características de los datos. Existen diferentes metodologías para estudiar las diferentes aproximaciones, como la búsqueda en cuadrícula o aleatoria. Sin embargo, se ha implementado optimización bayesiana, que ajusta un proceso gaussiano para modelar la función objetivo y equilibrar exploración y explotación. A medida que se obtienen más observaciones, mejora la distribución posterior, permitiendo explorar el espacio de parámetros de forma más eficiente y reduciendo el tiempo de búsqueda [38]. Una vez definida y evaluada la mejor aproximación para cada modelo, se procederá a su entrenamiento y aplicación sobre nuevos datos. Los resultados obtenidos se integrarán en **Edicom Analytics**, ofreciendo una visualización clara y accesible que facilitará su interpretación.

CAPÍTULO 6

Preparación y análisis de datos

La adecuada preparación del conjunto de datos es fundamental para garantizar un rendimiento óptimo de los modelos. En este capítulo se describen en detalle los procesos de creación, integración, transformación y procesamiento de los datos iniciales, explicando cómo fueron gestionados y analizados para maximizar su valor y utilidad en función de los objetivos del proyecto.

6.1 Extracción y creación de la base de datos

La información utilizada proviene de las trazas de transacciones EDI, recopiladas en la plataforma interna de la empresa. Estos registros contienen detalles capturados durante la monitorización del sistema, que capture todos los datos y eventos asociados a las transmisiones realizadas en la plataforma **Ediwin**. La extracción fue realizada a través de **Kibana**, una interfaz gráfica diseñada para explorar, visualizar y analizar la información almacenada en las bases de datos de **Elasticsearch**.

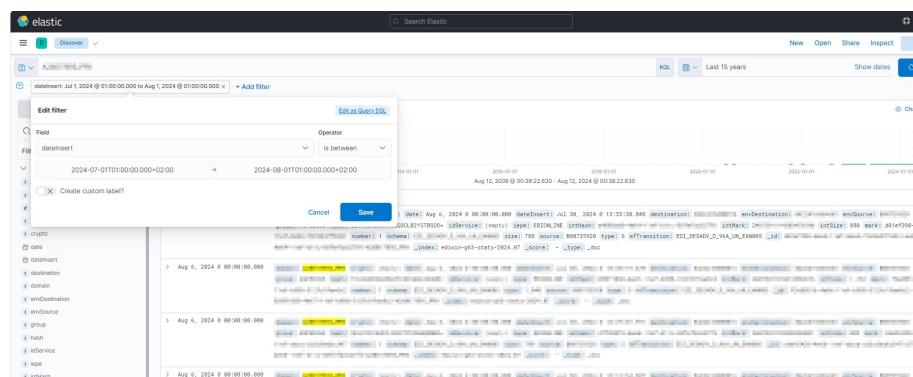


Figura 6.1: Vista y filtro en Kibana de la base de datos.

Debido a que el motor de búsqueda contiene bases de datos con información sensible sobre clientes y transacciones, existen restricciones de seguridad en cuanto a conexiones API, por lo que el proceso de extracción de datos se ha llevado a cabo de forma manual.

Como se muestra en la figura 6.1, este procedimiento consistió en filtrar la información de monitorización de un dominio específico para un cliente, aplicando además un filtro temporal para no superar el umbral de 20.000 observaciones. Esto se debe a que la extracción de datos se complica cuando se excede dicho límite. Una vez aplicado el filtro, se obtuvieron los resultados de la búsqueda y se descargó el informe en formato **.csv**.

Como resultado, se generaron 389 archivos **.csv** que recopilan la monitorización de los intercambios realizados en la plataforma del cliente durante el período comprendido entre el 1 de enero de 2022 y el 12 de diciembre de 2024. Posteriormente, todos los archivos fueron unificados en una única base de datos, que contiene un total de 7.117.213 observaciones y 25 variables. Estas variables son:

- **Id:** identificador del documento en la plataforma.
- **Index:** índice de los resultados en la base de datos.
- **Score:** función de puntuación predeterminada por Elastic search.

- **Crypto:** módulo de seguridad aplicado al mensaje.
- **Date:** fecha del documento (Solo la fecha, no la hora).
- **dateInsert:** fecha y hora a la que se insertó el mensaje en la plataforma.
- **Destination:** identificación del punto operacional destino.
- **Domain:** dominio de la plataforma del cliente donde se ha realizado la transacción.
- **envDestination:** identificación del Punto lógico destino.
- **envSource:** identificación del Punto lógico origen.
- **Group:** grupo en el que se encuentra la plataforma del cliente.
- **Hash/ intHash:** identificador hash del documento o del intercambio.
- **idService:** el id de servicio ejecutado.
- **Iepe:** módulo de comunicación aplicado al mensaje.
- **Mark/ intMark:** identificación interna del documento o del intercambio.
- **number:** unidades de mensajes transaccionados.
- **Schema:** interfaz del formato de mensaje transaccionado.
- **Size/ intSize:** tamaño del contenido o interacción del mensaje.
- **Source:** identificación del punto operacional origen.
- **Type:** tipo de flujo del mensaje (E: Entrada / S: Salida)
- **wfTransition:** valor del estado del mensaje en el Workflow de ElasticSearch.
- **_version:** versión del documento en ElasticSearch.

En conclusión, los datos extraídos monitorizan e identifican mensajes transaccionados en la plataforma del cliente, proporcionando información detallada sobre su origen, destino, tamaño, fecha, estado, tipo de flujo y las características técnicas asociadas a su procesamiento.

6.2 Transformación y análisis inicial

Una vez obtenido el conjunto de datos inicial, es crucial evaluar su calidad y estructura antes de proceder con el análisis. En esta etapa, se lleva a cabo un preprocesamiento que abarca la corrección de inconsistencias, la gestión de valores faltantes o duplicados, el tratamiento de valores atípicos y la eliminación de variables irrelevantes. La limpieza de datos es clave para mejorar su calidad mediante el tratamiento de valores ausentes, la suavización del ruido y la resolución de inconsistencias [32].

Las discrepancias en los datos pueden deberse a fallos en los sistemas de registro o inconsistencias en la recolección. Detectar y corregir estos problemas garantiza la fiabilidad del conjunto de datos y su óptimo aprovechamiento en modelos analíticos y de aprendizaje automático [32]. Es importante destacar que la presencia de valores ausentes no siempre indica un error, por lo que debe analizarse cada caso antes de imputarlos o eliminarlos.

Por último, es necesario realizar un análisis exploratorio preliminar para identificar patrones subyacentes en los datos y profundizar en su comprensión. A partir de los conocimientos obtenidos en esta etapa, se pueden agregar características diseñadas para capturar la variabilidad del conjunto de datos según los objetivos definidos.

6.2.1. Limpieza inicial

Inicialmente contamos con un conjunto de 7.117.213 observaciones y 25 variables. En esta sección, nos centraremos en asegurar la calidad de los datos previos a su análisis.

La plataforma **Ediwin** utiliza cuatro variables para identificar a los interlocutores, organizados en una libreta de contactos con estructura jerárquica para gestionar información empresarial. Esta clasificación distingue entre puntos operacionales, como almacenes o centros de distribución, y puntos lógicos, que representan empresas o divisiones que agrupan múltiples puntos operacionales. En total, se registran 513 valores únicos de puntos operacionales de destino, agrupados en 331 puntos lógicos. Para el origen, se identificaron 438 puntos operacionales distintos, organizados en 241 puntos lógicos únicos.

Aunque estos identificadores pueden ser códigos generados unilateralmente por el cliente, en algunos casos se utilizan los NIF para simplificar la identificación. Dado que es fundamental tomar las precauciones adecuadas para reducir los riesgos de reidentificación y divulgación de la información [20], tal y como se muestra en el listing 6.1, se ha aplicado un proceso de anonimización a las variables identificativas clave de los interlocutores, tanto de origen como de destino, en los mensajes entrantes y salientes de la plataforma, garantizando así la protección de su privacidad.

```

1. Obtener los valores únicos combinados de las columnas "source"/ "envSource" y
   "destination"/"envDestination" en el DataFrame.
   unique_source_destination = obtener valores únicos en el DataFrame.
2. Crear etiquetas anonimizadas únicas para cada valor en unique_source_destination.
   random_labels_unique_source_destination = generar etiquetas "entity1", "entity2", ...,
   "entityN", donde N es el número de elementos en unique_source_destination.
3. Mezclar las etiquetas anonimizadas aleatoriamente para anonimización.
4. Crear un diccionario de mapeo que asocia cada valor único de "source"/"envSource" y
   "destination"/"envDestination" con una etiqueta anonimizada.
   source_destination_to_anon = crear diccionario de mapeo entre
   unique_source_destination y random_labels_unique_source_destination.
5. Mapear las etiquetas anonimizadas a las columnas "source"/"envSource" y
   "destination"/"envDestination" en el DataFrame.
   para cada fila en el DataFrame:
       - Asignar el valor mapeado del origen usando el diccionario
         source_destination_to_anon.
       - Asignar el valor mapeado de destino usando el diccionario
         source_destination_to_anon.

```

Listing 6.1: anonimización de variables sensibles.

Una vez anonimizadas las identificaciones y con el objetivo de reducir el ruido generado por estas variables y facilitar un análisis más simple, directo y eficiente de las relaciones entre entidades, se creó una nueva variable, como se muestra en el listing 6.2, que representa los interlocutores de cada transacción como una tupla de origen y destino, dando lugar a 585 valores únicos.

```

Para cada fila en el DataFrame:
1. Ordenar los valores de las columnas "envSource" y "envDestination"
2. Unirlos con un guión bajo ("_") y asignar el resultado a la nueva columna
   "envSource_envDestination_transaction"

```

Listing 6.2: creación de una variable que representa los interlocutores involucrados en el intercambio del mensaje.

Continuamos en el listing 6.3, verificando la ausencia de registros duplicados en las transacciones, ya que durante la extracción o almacenamiento de los datos podrían haberse

generado duplicados por error. Para ello, comprobamos que cada transacción tenga un valor único en las variables que identifican los mensajes (`_id`, `hash` o `mark`).

```
Para cada fila en el DataFrame:
1. Identificar las columnas clave: "_id", "hash", "mark".
2. Verificar si existe otra fila con los mismos valores en estas columnas.
- Si se encuentra un duplicado:
  a. Mantener solo la primera aparición.
  b. Eliminar las filas duplicadas posteriores.
```

Listing 6.3: eliminación de observaciones duplicadas en el conjunto de datos utilizando variables de identificación.

Se han detectado 38.899 observaciones duplicadas, contando con un conjunto de 7.078.314 observaciones tras el filtrado.

Tras abordar los puntos iniciales, se realizó un análisis preliminar de las categorías de las variables para evaluar su alineación con los objetivos del proyecto y filtrar aquellas que pudieran generar redundancia o ruido. Como resultado, se descartaron las siguientes variables: `_index`, `_score`, `_type`, `date`, `hash`, `idService`, `intHash`, `intMark`, `number`, `wfTransition`, `mark`, `domain`, `group`. Estas corresponden a identificadores de transacciones, fechas comerciales o identificadores de plataformas del cliente que no contribuyen a la diferenciación de patrones clave ni a la separación de características relevantes en los datos.

Durante el análisis se identificaron valores anómalos en las variables `crypto` e `iepe` originados por datos faltantes en la base de datos, posiblemente debido a errores de recopilación o ausencia de información. Dado que no se dispone de detalles sobre la configuración de dichos protocolos, en el listing 6.4 se muestra cómo estos valores se reemplazarán por las categorías **NO CRYPTO** y **OTHER** para evitar que la falta de información sobre estas configuraciones afecte el análisis.

```
1. Definir los diccionarios de reemplazo para la variable crypto {"-": "NO CRYPTO", "": "NO CRYPTO", "NaN": "NO CRYPTO"} y para la variable iepe {"-": "OTHER"}.
2. Reemplazar los valores en la columna "crypto"/ "iepe" usando el diccionario correspondiente.
```

Listing 6.4: revisión de categorías para variables de módulos de seguridad y comunicaciones '`crypto`' e '`iepe`'.

Finalmente, en el listing 6.5, se corrige el formato de la columna `dateInsert`, convirtiéndola al tipo `datetime`. Además, se ajusta el formato de las variables numéricas `size` e `intSize` a tipo `float`.

```
1. Convertir la columna "dateInsert" de df A formato de fecha y hora usando el formato "%b %d, %Y @ %H:%M:%S.%f".
2. Eliminar las comas (",") en la columna "size"/"intSize".
3. Convertir la columna "size"/"intSize" a tipo float.
```

Listing 6.5: formato estandarizado para fechas y variables numéricas.

6.2.2. Análisis exploratorio inicial

Tras el preprocessado inicial, contamos con 7.078.314 observaciones y 13 variables anónimizadas. Para identificar patrones y características clave que permitan la comprensión de nuestro conjunto y el desarrollo de estrategias efectivas en las etapas posteriores del proyecto, realizaremos un análisis exploratorio inicial. Comenzaremos analizando las variables disponibles, especialmente las numéricas, para comprender sus distribuciones, rangos y posibles interacciones. Este análisis inicial nos permitirá obtener un conocimiento

profundo tanto de las distribuciones de las variables como de las relaciones entre estas [39], pudiendo aclarar su contribución al objetivo del proyecto. En cuanto a las variables categóricas, es posible visualizar su distribución según otras variables para exponer mejor sus relaciones [40].

Comenzaremos analizando la serie temporal de las variables `size` e `intSize` que miden el tamaño del contenido y del ensobrado del mensaje respectivamente. Se ha decidido priorizar el análisis de `size` para evaluar directamente el tamaño del mensaje, sin la influencia del ensobrado, el cual podría introducir sesgos al poder incluir contenido de varios mensajes relacionados y verse afectado por los protocolos de comunicación o seguridad.

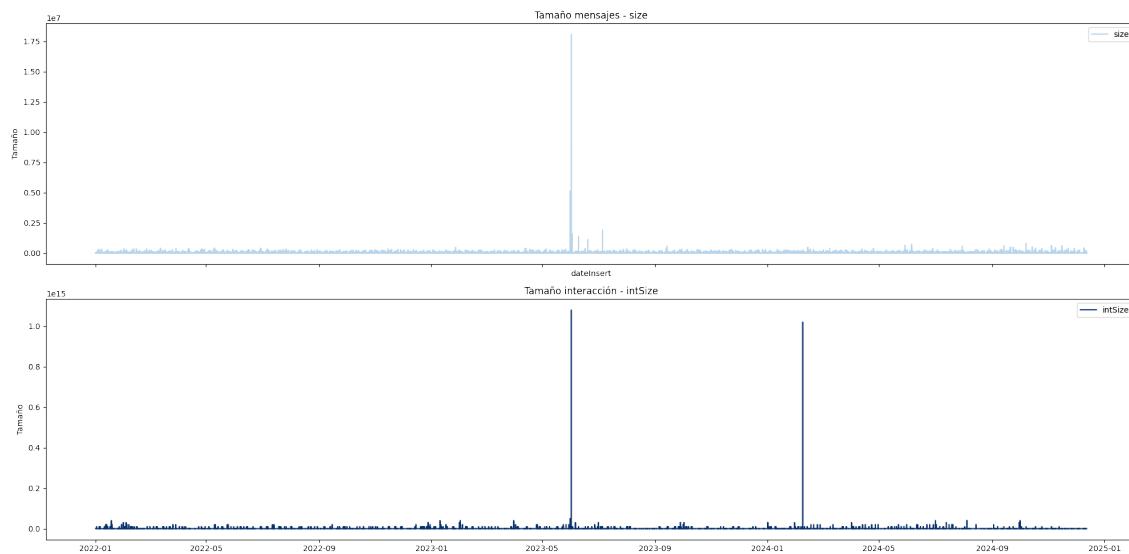


Figura 6.2: representación en series temporales del tamaño de contenido y del intercambio de mensajes.

En la figura 6.2, se observa que el tamaño del contenido y las interacciones de los mensajes presentan valores extremos que, aunque reales, no representan patrones regulares en las transacciones. Estos valores pueden afectar negativamente al modelado, distorsionando la distribución original de los datos y causando sobreajuste a los extremos reduciendo la capacidad de generalización. Es crucial por lo tanto tratar adecuadamente estos puntos atípicos antes de continuar con el análisis.

Para abordar este problema, dado que solo unas pocas observaciones presentan este problema, se ha optado por la eliminación. En el listing 6.6 se muestra como han sido identificadas estas observaciones, filtrando aquellas que superan el umbral del percentil 99.99 para las variables de tamaño. Resultando en la eliminación de 8 casos. Tras este proceso, contamos ahora con 7.078.306 muestras.

```

1. Calcular el percentil 99.99 para las columnas "size" e "intSize" y almacenar valores
   en percentil_99_size y percentil_99_intsize.
2. Filtrar filas donde "size" e "intSize" sean menores o iguales a sus respectivos
   percentiles.
df_filtrado = filtrar_filas(
    datos: df_unificado,
    condicion: (size menor o igual percentil_99_size) Y (intSize menor o igual
                percentil_99_intsize)
)

```

Listing 6.6: tratamiento de valores atípicos en las variables 'size' e 'intSize'.

Tras el filtrado anterior, los gráficos de series temporales de la figura 6.3, revelan patrones más regulares y analizables. Aunque no se distingue una tendencia clara para el tamaño del contenido de los mensajes, debido a la naturaleza irregular de las transacciones, se observan picos que podrían estar relacionados con la diferencia entre períodos de baja actividad, en los que no se registraron valores, y aquellos en los que sí se detectaron. A pesar de esta variabilidad, es posible delimitar los rangos que agrupan la mayoría de los tamaños máximos. Además, se observa una tendencia general al aumento de los valores, acompañada de un incremento en la frecuencia de aparición de mensajes de mayor tamaño.

Por otro lado, el tamaño de los intercambios presenta un rango mucho más amplio. Si bien los valores más elevados tienden a seguir patrones más definidos y muestran una mayor homogeneidad, no se identifica una tendencia clara ni un patrón específico.

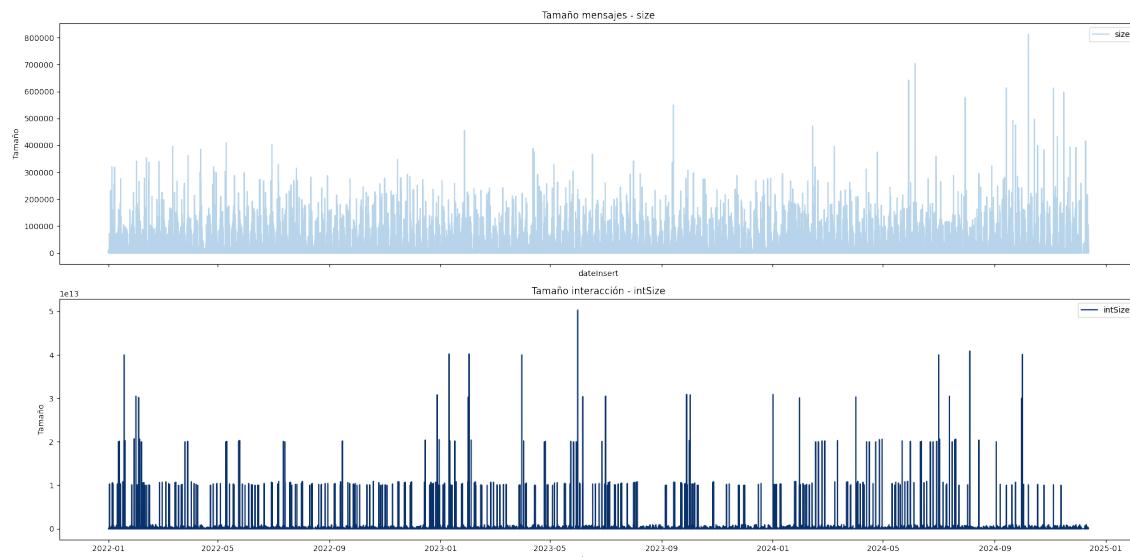


Figura 6.3: representación en series temporales del tamaño del contenido e intercambio de mensajes una vez aplicado el filtrado.

En la figura 6.4, la comparación de la distribución de ambas variables mediante gráficos BoxPlot revela que, aunque intSize abarca un rango de valores mucho más amplio, alcanzando hasta 50.000.000.000.000 GB, mientras que el tamaño del contenido de los mensajes llega a un máximo de 800.000 GB, ambas variables presentan una menor frecuencia de valores atípicos a medida que los tamaños aumentan, concentrando la mayoría de las observaciones en valores cercanos a 0.

No obstante, la distribución de los valores extremos en los mensajes ensobradados sigue un patrón más escalonado, con valores atípicos agrupados en rangos específicos y una menor frecuencia en comparación con la distribución de los valores extremos del contenido, que exhibe una mayor dispersión a lo largo de todo su rango.

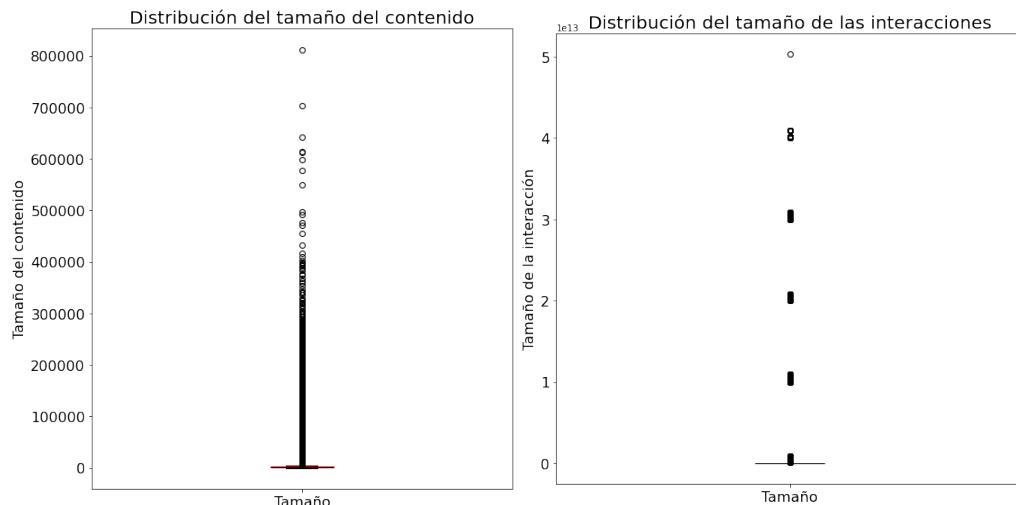


Figura 6.4: visualización mediante diagramas de caja de la distribución del contenido y del flujo de mensajes, posterior al tratamiento de valores extremos.

Como se muestra en el listing 6.7, se ha añadido una nueva variable categórica, `size_cat`, derivada del tamaño del contenido, que clasifica los valores en cuatro categorías: **Low**, **Medium_Low**, **Medium_High** y **High** definidas según los percentiles de la variable `size`, de esta forma se facilita una agrupación más clara.

1. Dar valor a `q1_size`, `q2_size` y `q3_size` con los percentiles 25, 50, 75 de la columna "size" respectivamente.
2. Dar valor `bins_size` A [mínimo de "size", `q1_size`, `q2_size`, `q3_size`, máximo de "size"]
3. Fijar etiquetas `["Low", "Medium_Low", "Medium_High", "High"]`
Crear columna `"size_cat"` fijando las etiquetas definidas a los rangos de bins definidos.

Listing 6.7: categorización de la variable 'size' mediante intervalos derivados de su distribución cuartílica.

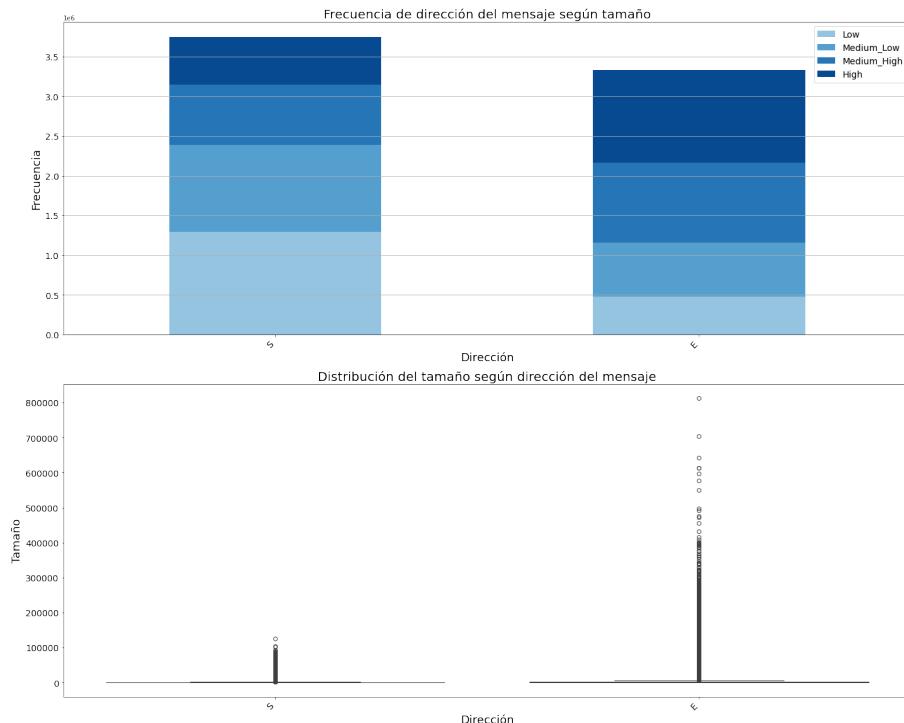


Figura 6.5: análisis de frecuencia y distribución del tamaño según la dirección del mensaje.

Al analizar la frecuencia de las direcciones según la categoría de tamaño, se observa que los mensajes entrantes (clase E) presentan una mayor proporción de tamaños elevados, mientras que los mensajes salientes (clase S) se concentran principalmente en categorías de menor tamaño. Esto sugiere que los mensajes recibidos tienden a ser, en general, más complejos y voluminosos.

Esta tendencia también se confirma en el BoxPlot de la figura 6.5, donde, a pesar de que la media es similar en ambas direcciones, la distribución de los mensajes entrantes se inclina hacia valores más altos.

De acuerdo con la figura 6.6, dónde se diferencian dos gráficas de series temporales del tamaño según la dirección de los mensajes. A pesar de no observar un patrón claro, es evidente la diferencia en los tamaños entre ambas direcciones: los valores máximos para los mensajes salientes oscilan entre 50.000 y 90.000, mientras que para los entrantes se encuentran en un rango significativamente mayor, entre 200.000 y 400.000.

Se destaca una mayor estabilidad en los valores para las transacciones salientes, donde se aprecian menos picos identificando bloques de mensajes con tamaños similares y uniformes en comparación con el flujo contrario con más variabilidad presente.

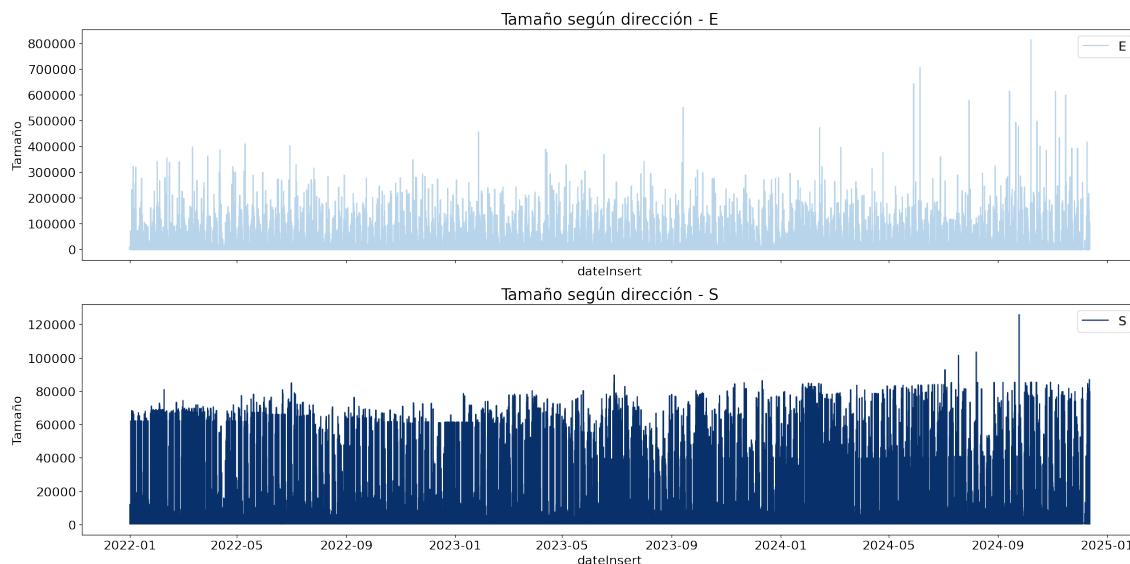


Figura 6.6: representación en serie temporal del tamaño del mensaje según su dirección.

Estudiamos a continuación la variable `schema` que refleja los diversos tipos de mensajes transaccionados por el cliente a través de la plataforma, los cuales aportan información relevante sobre su perfil. En la figura 6.7 se muestra la frecuencia de aparición de estos tipos de mensajes destacando dos clases con frecuencias extremadamente bajas: **EDI_INVOIC_D_96A_UN_EAN008** que aparece en dos ocasiones y **EDI_GENRAL_2_000_00_000000** que solo se envía una vez. Este comportamiento sugiere que estos datos no corresponden a transacciones reales ni periódicas del cliente. Conservarlos en el conjunto solo añadiría ruido y afectaría la calidad del análisis. Por ello, hemos decidido eliminarlos.

Los mensajes más frecuentes son pedidos (**ORDERS**) y reportes de estado de pedidos (**OSTRPT**), seguidos por facturas (**INVOIC**) y respuestas de pedido (**ORDRSP**). Los menos comunes son los albaranes (**DESADV**) y mensajes generales de texto (**GENERAL**). En cuanto a la dirección, el cliente recibe albaranes, pedidos y facturas, y envía pedidos, respuestas a pedidos y reportes de estado, lo que sugiere un rol de intermediario logístico y comercial, actuando como distribuidor, minorista o mayorista dentro de la cadena de suministro, con flujos significativos de compra y venta.

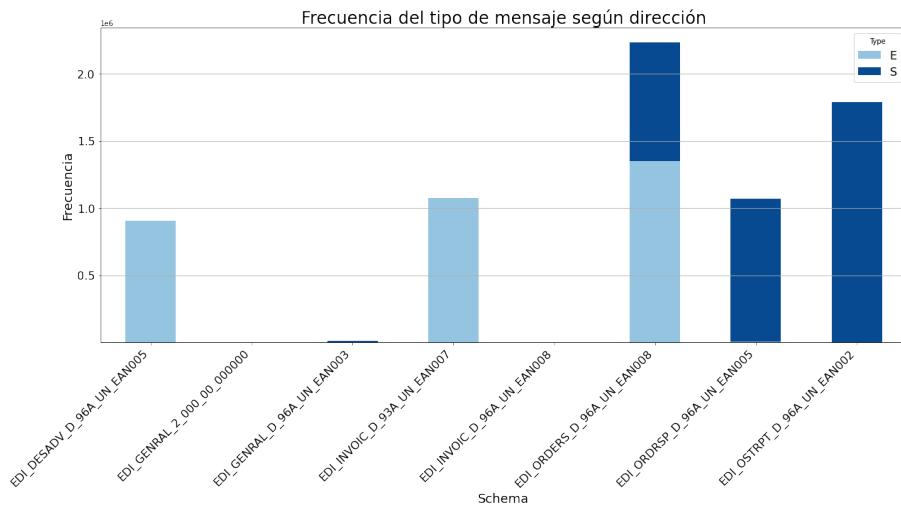


Figura 6.7: análisis de frecuencia y distribución del tipo de mensaje según su dirección.

La mayor frecuencia de **ORDERS** y **OSTRPT** destaca la importancia de la gestión y seguimiento de pedidos, reflejando la necesidad de sincronización constante con los socios comerciales y el manejo de grandes volúmenes de interacciones. **INVOIC** y **ORDRSP**, aunque menos frecuentes, refuerzan la finalización de operaciones mediante facturas y confirmaciones de pedido, asegurando cierres claros en las transacciones. La baja presencia de **DESADV** sugiere que las notificaciones de envío, o bien no son críticas en su operativa, están gestionadas por otros actores dentro de la cadena de suministro, o se emiten de forma agrupada, consolidando varios pedidos. Finalmente, **GENRALS** parecen emplearse sólo en comunicaciones no críticas o situaciones excepcionales, como errores o avisos por lo que tiene poca aparición en el flujo.

En conjunto, el perfil operacional del cliente revela una gestión eficiente de grandes volúmenes de información, con un alto nivel de automatización característico de empresas que operan como mayoristas en entornos B2B o conectan fabricantes y consumidores finales en modelos de distribución.

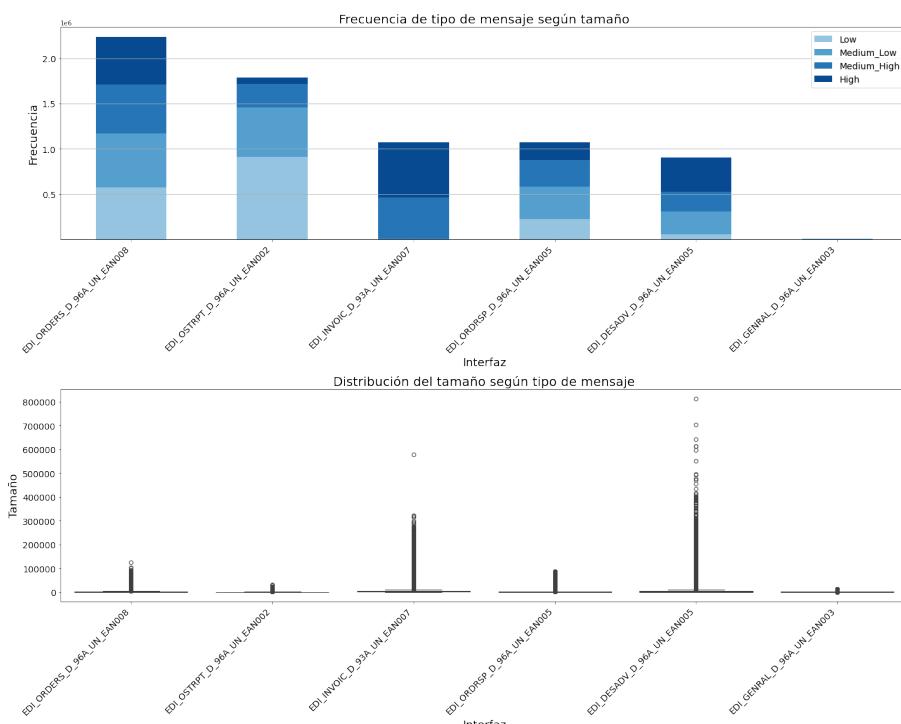


Figura 6.8: análisis de frecuencia y distribución del tamaño del mensaje según su tipo.

La figura 5.8 combina gráficos de frecuencia y BoxPlot para analizar la distribución del tamaño segmentada por tipo de mensaje, ofreciendo una visión integral de las diferencias entre ellos.

Los mensajes de tipo **ORDERS** presentan una distribución relativamente equilibrada en todas las categorías de tamaño, lo que sugiere una variabilidad considerable en su contenido. En cambio, los mensajes **INVOIC** y **DESADV** muestran patrones similares, caracterizados por una clara tendencia hacia tamaños grandes. Estos tipos de mensaje no solo tienen una mayor dispersión en su distribución, como lo evidencia el BoxPlot, sino que también exhiben bigotes que se extienden hacia valores extremos y una cantidad significativa de valores atípicos, especialmente pronunciada en los albaranes.

Por otro lado, los mensajes **ORDRSP** y **OSTRPT** destacan por concentrar sus tamaños en torno a valores más pequeños, con distribuciones más compactas y menos dispersión. Aunque también se identifican valores atípicos, estos son menos frecuentes y menos extremos que en los casos de **INVOIC** y **DESADV**.

En síntesis, los mensajes **INVOIC** y **DESADV** tienden a contener información más detallada y extensa, lo que explica sus tamaños mayores y su dispersión más amplia. Por el contrario, los mensajes **ORDRSP** y **OSTRPT**, al reflejar transacciones más simples, como respuestas y reportes, se caracterizan por tamaños más pequeños y uniformes. Los mensajes **ORDERS**, al situarse entre ambos extremos, muestran una variabilidad, posiblemente debido a la diversidad de órdenes procesadas.

Al analizar el gráfico de series temporales de cada tipo de mensaje de la figura 5.9, se observa que, hasta 2023, los mensajes de tipo **ORDERS** y **ORDRSP** mantienen una estabilidad inusitada en su tamaño. Sin embargo, a partir de ese año, comienza a notarse una mayor aleatoriedad, lo que podría ser consecuencia de la inclusión de otros flujos de negocio.

A pesar de la variabilidad mencionada, el rango del tamaño de los mensajes sigue siendo relativamente constante. Además, se destacan tamaños inusualmente grandes para el mensaje **GENRAL** alrededor de enero de 2023, aunque este comportamiento parece ser un caso aislado y no se repite posteriormente. En cuanto a los demás tipos de mensajes, se observan picos en el tamaño de manera aparentemente aleatoria, sin patrones claros que faciliten su interpretación.

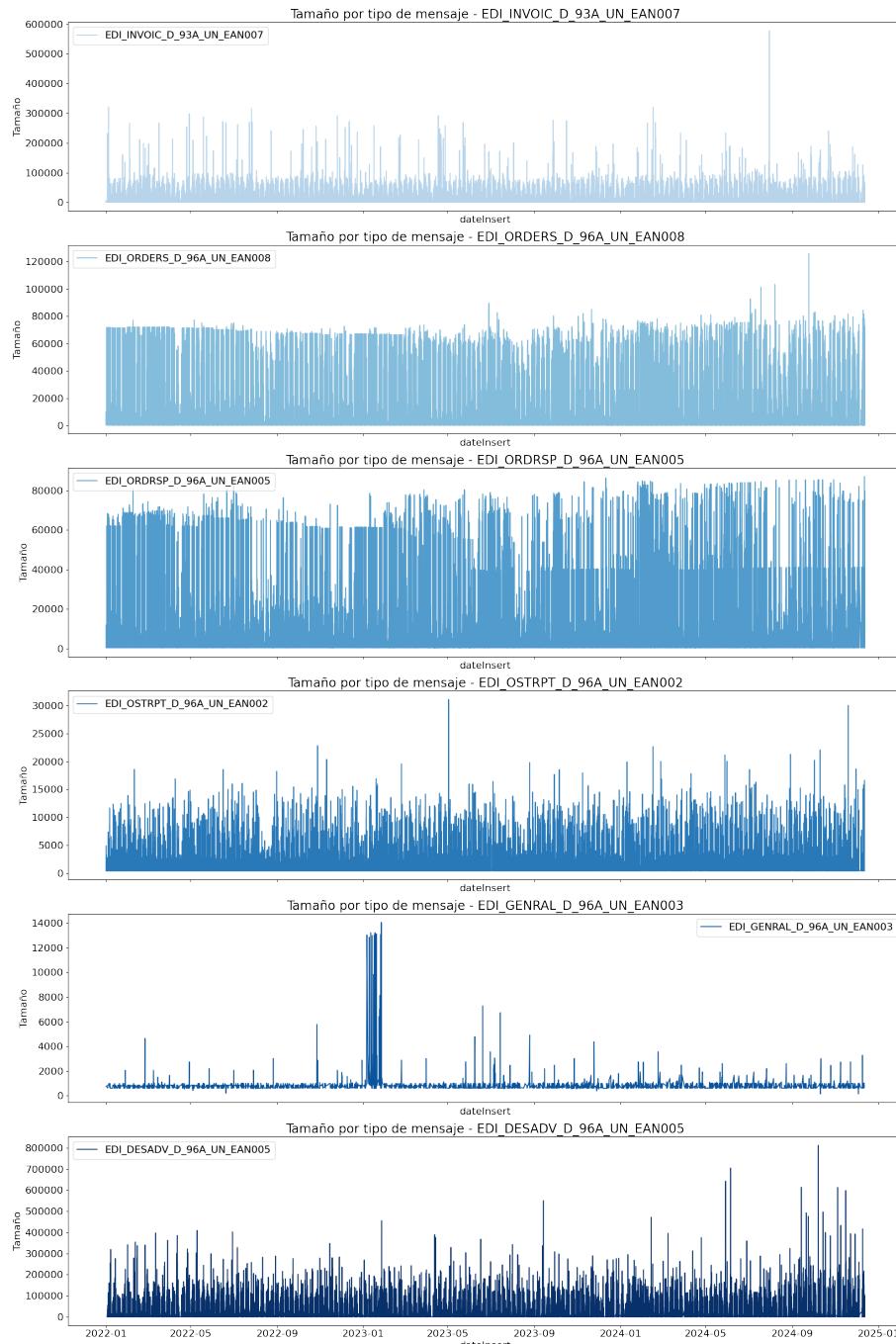


Figura 6.9: representación en serie temporal del tamaño del mensaje según su tipo.

Al analizar el gráfico de barras de la figura 6.10 correspondiente a la variable `iepe`, que representa el protocolo de comunicaciones utilizado en las transacciones, podemos ver para el protocolo más frecuente, **EBIONLINE**, la presencia de una distribución relativamente equilibrada entre mensajes entrantes y salientes, con una leve predominancia en estos últimos.

Cabe mencionar, que se han excluido las clases **OTHER** y **EBIONLINE2** debido a su baja frecuencia, con solo 187 y 101 apariciones respectivamente, y únicamente presentes en mensajes salientes.

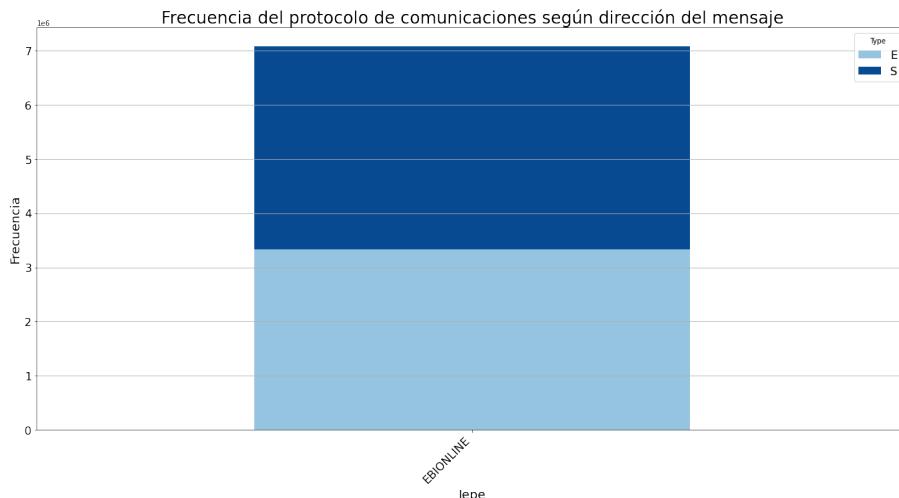


Figura 6.10: análisis de frecuencia y distribución del protocolo de comunicaciones aplicado al mensaje según su dirección.

El gráfico de series temporales en la figura 5.11 revela que las transmisiones del tipo **OTHER**, aunque de muy baja frecuencia, presentan un pico en tamaño alrededor de abril de 2023. Por su parte, la comunicación **EBIONLINE2** fue activa principalmente entre mayo de 2022 y enero de 2023, con tráfico esporádico en 2024 tras cesar su actividad regular. Ambos tipos de comunicación parecen atípicos y podrían introducir ruido o sesgo en el análisis, probablemente por su uso temporal en procesos de migración, pruebas con clientes externos o interacciones puntuales. Dado su carácter excepcional, es posible que no aporten diferenciación relevante del comportamiento transaccional.



Figura 6.11: representación en serie temporal del tamaño del mensaje según el protocolo de comunicaciones aplicado.

Analizando la variable *crypto*, que representa el protocolo de seguridad aplicado, la figura 6.12 revela un comportamiento esporádico del protocolo **SMOJERACUN**, con solo 64 intercambios registradas en abril de 2022, tras lo cual dejó de usarse. Dado su carácter aislado y para evitar introducir ruido innecesario, estas transacciones fueron eliminadas, ya que el resto de clases aportan mayor diferenciación al análisis.

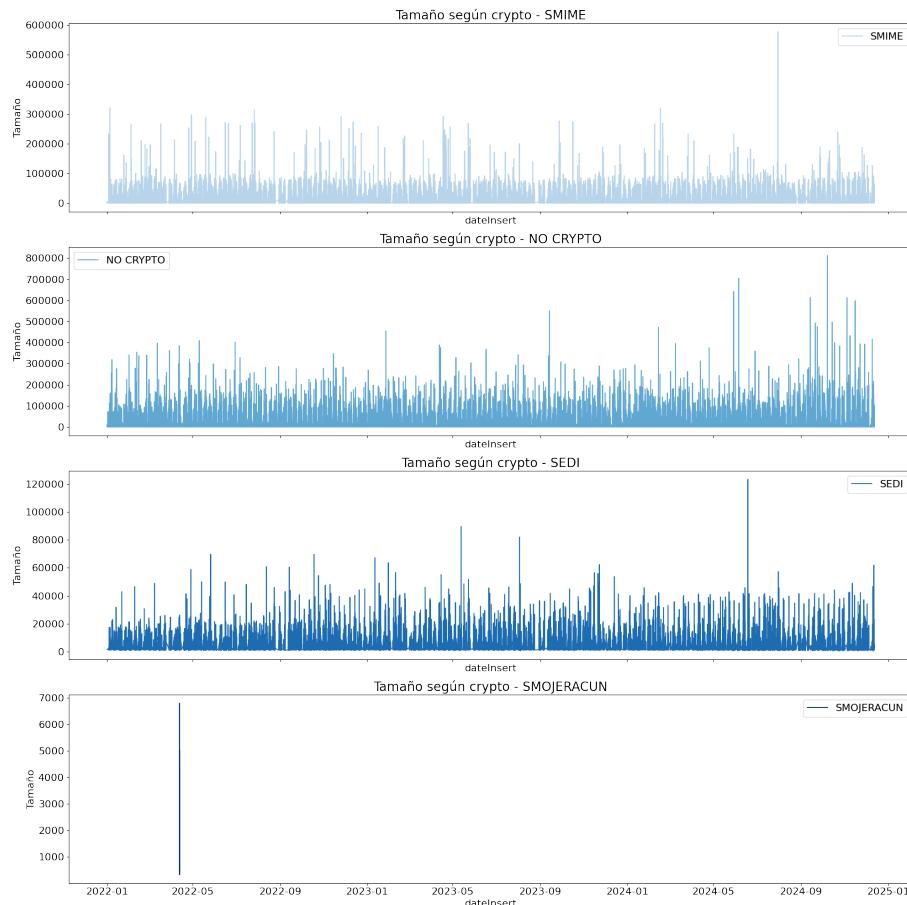


Figura 6.12: representación en serie temporal del tamaño del mensaje según el protocolo de seguridad aplicado.

Al analizar la distribución del tamaño de los mensajes según su tipo mediante gráficos combinados de frecuencia y BoxPlot en la figura 6.13, se observa que el protocolo más frecuente es **NO CRYPTO**, asociado a mensajes con tamaños generalmente menores pero con una alta variabilidad.

Por otro lado, **SMIME**, aunque menos frecuente, muestra una distribución de tamaños más amplia, con un cuartil superior notablemente elevado, lo que indica que los tamaños de sus mensajes tienden a situarse por encima de la media. Finalmente, **SEDI**, el protocolo menos común, se caracteriza por transacciones de mayor tamaño y presenta el menor número de valores atípicos, lo que sugiere una distribución más homogénea en comparación con los otros protocolos.

En conjunto, mientras **NO CRYPTO** y **SMIME** comparten una mayor cantidad de valores atípicos, estos son más marcados en **SMIME** debido a su concentración en tamaños elevados. La visualización combinada revela que las diferencias en frecuencia y rango entre los protocolos están estrechamente relacionadas con la naturaleza y propósito de cada tipo de mensaje.

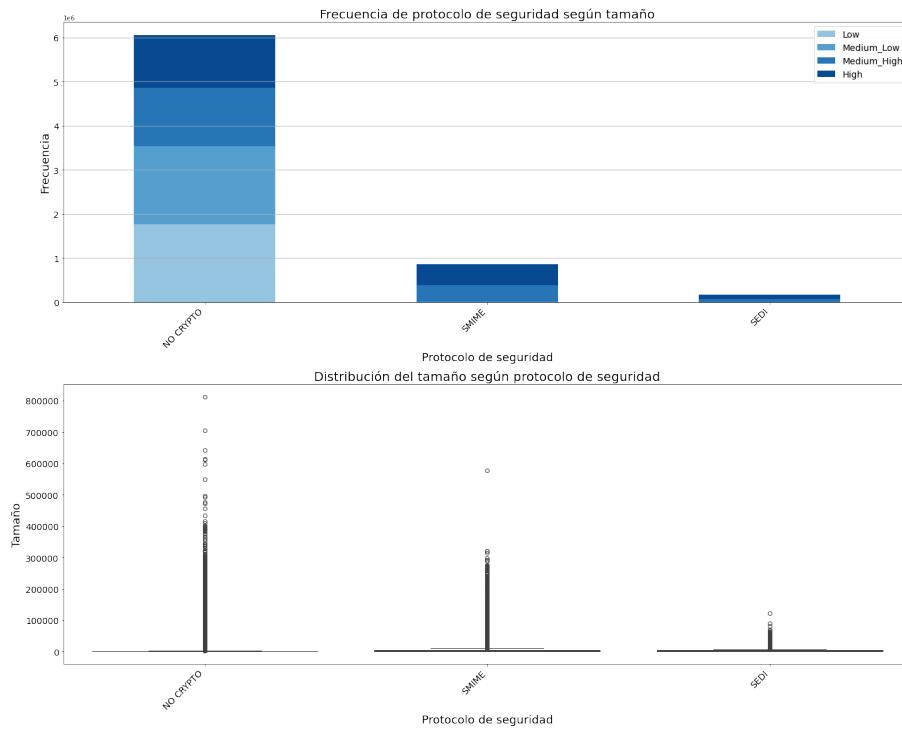


Figura 6.13: análisis de frecuencia y distribución del tamaño del mensaje según su tipo.

Los gráficos de barras en la figura 6.14 revelan una distribución distintiva en la frecuencia de aparición de los mensajes según su dirección y tipo. Los protocolos de seguridad **SEDI** y **SMIME** se asocian exclusivamente con mensajes entrantes y están presentes únicamente en mensajes de tipo **INVOIC**. En contraste, los mensajes sin seguridad se encuentran tanto en dirección de entrada como de salida y abarcan otros tipos de contenido distintos a facturas. Esta diferenciación sugiere que la implementación de protocolos de seguridad está estrechamente vinculada al tipo de mensaje y a su dirección, con un enfoque específico en la protección de transacciones relacionadas con facturación.

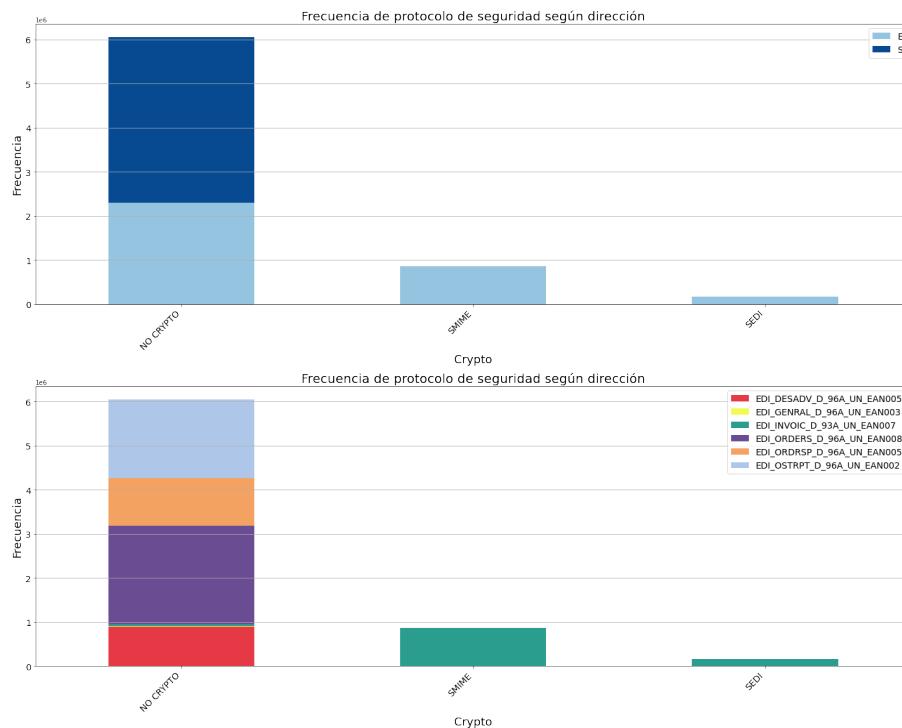


Figura 6.14: análisis de frecuencia y distribución del protocolo de seguridad aplicado a los mensajes segun su dirección y tipo.

El análisis del origen y destino de los intercambios es esencial para comprender los patrones de comportamiento de los clientes. Sin embargo, las variables relacionadas con estos elementos `source`, `destination`, `envSource`, `envDestination` pueden generar un nivel significativo de ruido debido a la diversidad de categorías. En total, se identifican 513 destinos únicos y 331 destinos finales, junto con 438 orígenes distintos y 241 orígenes finales únicos. Aunque estas variables son útiles para detectar patrones, su alta cardinalidad dificulta el análisis y limita su efectividad en la segmentación.

En la figura 11.1 del anexo se presentan los diez interlocutores más frecuentes en los puntos operacionales de origen y destino de los mensajes. Asimismo, en la figura 11.2 se ilustra la frecuencia de los interlocutores lógicos, destacando a `envEntity121` como el más frecuente y principal interlocutor de nuestro cliente. No obstante, también se observa una alta frecuencia de mensajes asociados a `envEntity349`, otro interlocutor destacado.

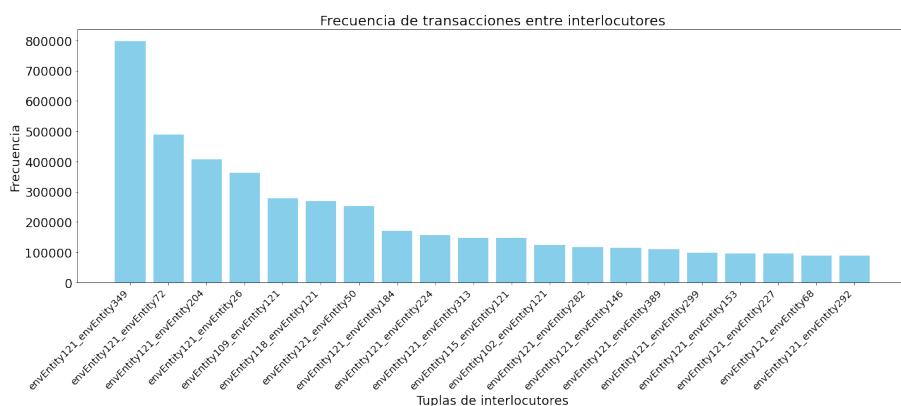


Figura 6.15: análisis de frecuencia y distribución de los mensajes según las tuplas de interlocutores involucrados.

Para obtener una perspectiva más clara de los patrones transaccionales, se ha creado la variable `envSource_envDestination_transaction`, que agrupa las tuplas de origen y destino final. El análisis de la frecuencia de estas tuplas, como se muestra en la figura 6.15, revela que la interacción más frecuente entre origen y destino corresponde a aquella entre nuestro cliente y el interlocutor 349, seguida por el interlocutor 72, con frecuencias aproximadas de 800.000 y 500.000, respectivamente.

Debido a la elevada cantidad de clases, persiste un considerable nivel de ruido. Por esta razón, se ha decidido agrupar las clases menos frecuentes, dado que muchas tuplas presentan una frecuencia baja. Esto permite centrar el análisis en las interacciones más representativas del comportamiento de los clientes.

Tras aplicar esta transformación, como se muestra en el listing 6.8, contamos con 53 categorías únicas, siendo una de estas **OTHER** que representa aquellos orígenes y destinos con menor frecuencia de aparición.

1. Establecer umbral de 30 000 apariciones para determinar las categorías frecuentes
2. Contar la frecuencia de cada categoría en la columna
`"envSource__envDestination__transaction"`
3. Identificar las categorías cuya frecuencia es mayor o igual al umbral y almacenar en una lista
4. Crear una nueva columna `"envSource__envDestination__transaction2"` donde:
 - Si la categoría está en la lista de clases que superan el umbral, se mantiene igual.
 - Si no, se reemplaza por clase **"OTHER"**.

Listing 6.8: Agrupación de clases poco frecuentes para 'envSource_envDestination_transaction'.

En la figura 6.16 se presentan gráficos que muestran la frecuencia de las interacciones, segmentadas por dirección y tipo de mensaje. A partir del análisis de los patrones subya-

entes, es posible identificar distintos perfiles de interacción.

Por un lado, se observa un perfil centrado en la recepción de órdenes y el envío de respuestas o estados, como ocurre en las interacciones con el interlocutor 26, donde se reciben mensajes del tipo ORDERS y se responden con ORDRSP u OSTRPT. Otro perfil relevante corresponde al envío de pedidos y la recepción de facturas o albaranes, evidente en puntos como los interlocutores 153, 113 y 282, donde se reciben documentos logísticos mientras se emiten pedidos. Finalmente, también se detecta un grupo de clientes, como los identificados con los códigos 159, 202, 76 y 68, cuyas interacciones están marcadas principalmente por un flujo de pedidos entrantes.

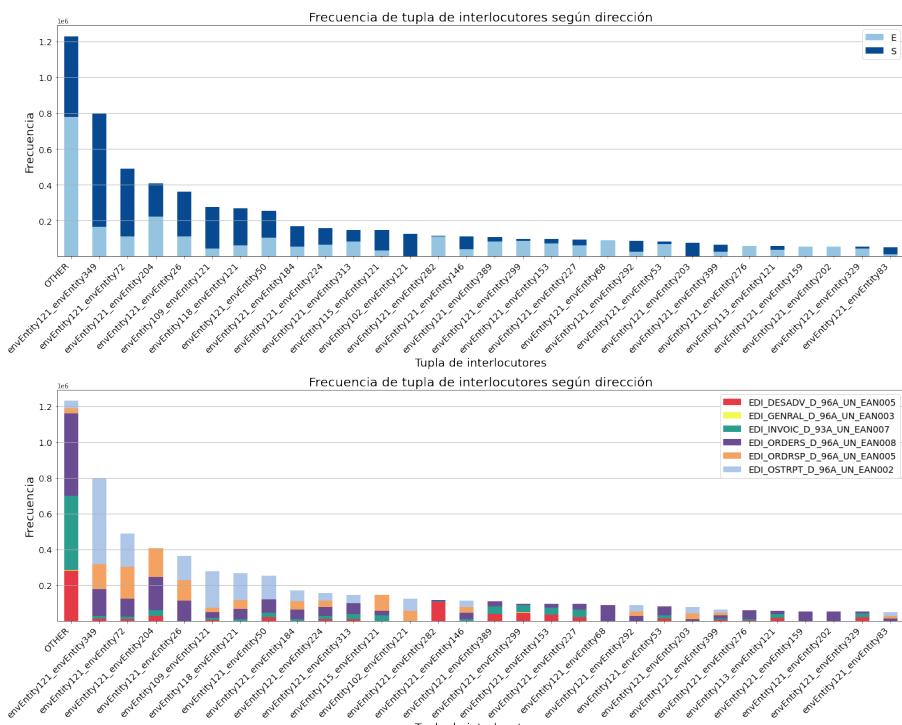


Figura 6.16: análisis de frecuencia y distribución de las tuplas de interlocutores involucrados en mensajes según su dirección y tipo.

En la figura 6.17 se analiza la distribución del tamaño de los mensajes para las 30 tuplas de interlocutores más frecuentes. Este análisis combina la información del gráfico de frecuencias y el boxplot, proporcionando una visión más detallada de los patrones según las características de los interlocutores.

Se observa que las transacciones con el cliente 349 presentan una mayor proporción de mensajes de menor tamaño, lo que contrasta con los mensajes asociados al cliente 204, donde predominan los tamaños más elevados. Por su parte, las interacciones con el cliente 72 muestran una distribución más equilibrada en las proporciones de los tamaños de los mensajes.

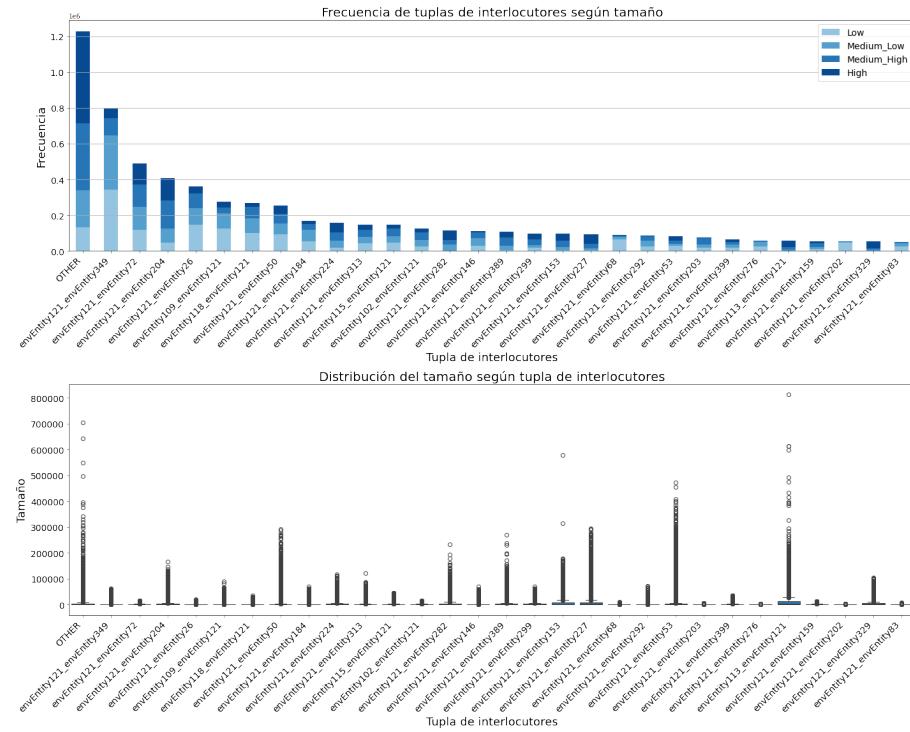


Figura 6.17: análisis de frecuencia y distribución del tamaño de mensaje según las tuplas de interlocutores involucrados.

En el boxplot, aunque todas las tuplas comparten una mediana homogénea, se identifican patrones específicos. Las transmisiones vinculadas a las entidades **93**, **232** y **193** destacan por su alta variabilidad y un número significativo de valores atípicos. Además, se observa que las tuplas con las entidades **68**, **115**, **232**, **390**, **193**, **389**, **159** y **202** tienen un rango intercuartílico superior más amplio, indicando la frecuencia de valores elevados. Este efecto es particularmente notable en las interacciones con la entidad **68**, donde los tamaños más grandes son especialmente frecuentes.

Tras el análisis, hemos adquirido una comprensión más profunda de los datos y sus relaciones. Se ha decidido eliminar las variables `envSource`, `envDestination`, `source`, `destination` y `iepe`, debido a su mínima contribución a la diferenciación de patrones o a la presencia de otras variables que ya impliquen estas características, lo que justifica su exclusión para reducir el ruido y mejorar la calidad de los datos. Además, se han identificado patrones relevantes en las transacciones de los interlocutores, así como en los perfiles de flujos EDI y configuraciones, que serán clave para discernir características en el conjunto de datos.

Finalmente, contamos con 7.078.242 observaciones y 8 variables entre las que se encuentran 5 categóricas `crypto`, `envSource_envDestination_transaction2`, `schema`, `type` y `size_cat`) dos variables numéricas `size`, `intSize` y una variable de fecha `dateInsert`. Este conjunto final servirá como base para alimentar los modelos de predicción y clusterización, con el objetivo de alcanzar las metas definidas en la sección 2.2. No obstante, aunque ambos enfoques partirán del mismo conjunto de datos, será necesario adaptarlo a las particularidades de cada método para maximizar su efectividad y asegurar el éxito.

CAPÍTULO 7

Modelos de clustering

La identificación de patrones de comportamiento en grandes volúmenes de datos transaccionales es esencial para comprender la dinámica del sistema y personalizar los servicios ofrecidos. En este proyecto, mediante el uso de algoritmos de agrupamiento, se busca organizar las operaciones del entorno de un cliente según sus características, lo que facilita su diferenciación en grupos significativos y mejora la comprensión de las transmisiones. Esta segmentación permite identificar particularidades tanto del cliente como de sus operaciones, y revelar dinámicas subyacentes que no son evidentes a simple vista, como hábitos de uso asociados a patrones temporales o atributos específicos de los mensajes, tales como la dirección, el tamaño, el módulo de seguridad, los interlocutores involucrados, entre otros. Todo ello genera insights valiosos para el análisis y la toma de decisiones estratégicas.

Este capítulo se centra en la identificación de características relevantes y en la potenciación de los datos para maximizar la efectividad de los modelos de clusterización, así como su posterior aplicación, con especial énfasis en los algoritmos de K-Means y HDBSCAN, seleccionados por su equilibrio entre eficiencia, escalabilidad y capacidad de adaptarse a estructuras de datos complejas. El objetivo final de su implementación es optimizar la asignación de recursos, fortalecer la inteligencia de negocio y respaldar decisiones informadas en un entorno caracterizado por una alta variabilidad transaccional.

7.1 Agregación de características y preparación de datos para clustering

En esta sección se describen las estrategias empleadas para la creación y selección de características a partir de los datos disponibles, con el propósito de preparar la información para su aplicación en modelos de clusterización. El enfoque se centra en identificar e incorporar nuevas características para aumentar la variabilidad del conjunto de datos y que permitan segmentar el flujo según patrones operacionales.

Con el objetivo de ampliar la representación del conjunto, se añadieron variables orientadas a cubrir las siguientes áreas:

1. Medidas estadísticas básicas sobre el tamaño y su dinámica

Con el objetivo de capturar variaciones en el tamaño de los mensajes para identificar distintos tipos de transmisiones, se calculan estadísticas básicas como la Media, desviación estándar, Z-score del tamaño de los mensajes en el listing 7.1.

```
1. Calcular el promedio del tamaño:  
    mean_size = promedio de la columna "size"  
2. Calcular la desviación estándar del tamaño:  
    std_dev_size = desviación estándar de la columna "size"  
3. Calcular la desviación del tamaño con respecto al promedio para cada fila:  
    Para cada fila en la columna "size":  
        size_deviation_from_mean = valor de "size" - mean_size  
4. Calcular el puntaje Z para cada fila en la columna "size":  
    Para cada fila en la columna "size":  
        size_z_score = (valor de "size" - mean_size) / std_dev_size
```

Listing 7.1: análisis estadístico de la variable size utilizando medidas de tendencia central y dispersión para evaluar su comportamiento y variabilidad.

Por otro lado, en el listing 7.2 se muestra la creación de métricas que miden las tendencias dinámicas como la diferencia en el tamaño entre mensajes consecutivos (size_rate_of_change) o la proporción acumulada del tamaño (cumulative_size_ratio), que mide el peso relativo del tamaño de un mensaje respecto al total.

```

1. Calcular el total acumulado del tamaño:
    cumulative_size = suma acumulativa de los valores en la columna "size"
2. Calcular el total global del tamaño:
    total_size = suma total de los valores en la columna "size"
3. Calcular la proporción acumulada del tamaño para cada fila:
    Para cada fila en la columna "size":
        cumulative_size_ratio = cumulative_size / total_size
4. Calcular la tasa de cambio del tamaño para cada fila:
    Para cada fila en la columna "size" (excepto la primera):
        size_rate_of_change = diferencia con el valor anterior ("size") /
            valor actual ("size")

```

Listing 7.2: cálculo de métricas acumulativas y tasas de cambio de la variable 'size' para analizar su evolución y distribución.

2. Dimensiones temporales

Con el objetivo de identificar patrones estacionales, representando la naturaleza cíclica de las variables temporales, capturar patrones referentes a festivos, días laborables y horas de mayor tránsito.

En el listing 7.3 se calcula la densidad transaccional o tasa de eventos por unidad de tiempo (transaction_density), es decir, el número de mensajes por segundo.

```

1. Calcular la diferencia temporal entre fechas consecutivas:
    Para cada fila en la columna "dateInsert" (excepto la primera):
        time_difference = diferencia con la fecha anterior en segundos
2. Calcular la densidad de transacciones:
    Para cada fila en la columna "dateInsert" (excepto la primera):
        transaction_density = 1 / time_difference

```

Listing 7.3: cálculo de variables que explican la densidad transaccional mediante la diferencia temporal entre transacciones y su tasa de ocurrencia.

En el listing 7.4 se calculan las variables cíclicas obteniendo el seno y coseno de la hora, día de la semana, día del mes y semana del año. También se agregan variables binarias indicando si la observación ha sido tomada durante el fin de semana o no.

```

1. Crear variables cíclicas para la hora del día:
    Para cada fila en "dateInsert":
        hour_sin = sin(2 * π* (hora de "dateInsert") / 24)
        hour_cos = cos(2 * π* (hora de "dateInsert") / 24)
2. Crear variables cíclicas para el día de la semana:
    Para cada fila en "dateInsert":
        day_of_week_sin = sin(2 * π* (día de la semana de "dateInsert") / 7)
        day_of_week_cos = cos(2 * π* (día de la semana de "dateInsert") / 7)
3. Crear variables cíclicas para el mes:
    Para cada fila en "dateInsert":
        month_sin = sin(2 * π* (mes de "dateInsert") / 12)
        month_cos = cos(2 * π* (mes de "dateInsert") / 12)
4. Crear variables cíclicas para la semana del año:

```

```

Para cada fila en "dateInsert":
    week_of_year_sin = sin(2 * π* (semana ISO del año de "dateInsert") /
                           12)
    week_of_year_cos = cos(2 * π* (semana ISO del año de "dateInsert") /
                           12)

5. Identificar si el día es fin de semana:
    Para cada fila en "day_of_week":
        Si el día de la semana es 5 (sábado) o 6 (domingo):
            is_weekend = 1
        En caso contrario:
            is_weekend = 0

```

Listing 7.4: cálculo de variables cíclicas temporales.

Asimismo, se obtienen los ratios de tamaño por hora, día, mes y semana del año, como se puede observar en el listing 7.5, además del Z-score del tamaño agrupado por estas categorías (por ejemplo hour_size_zscore) y la proporción acumulativa del tamaño en función de las categorías temporales (cumulative_<category>_ratio).

```

1. Calcular la proporción del tamaño por hora:
    Agrupar los datos por hora de "dateInsert" y contar el número de filas en
    cada grupo:
        hour_group_count = total de filas en cada "hour"
    Dividir el conteo por el número total de filas en el DataFrame:
        Para cada valor en hora de "dateInsert":
            hour_size_ratio = hour_group_count / total_filas

2. Calcular la proporción del tamaño por día de la semana:
    Agrupar los datos por día de la semana de "dateInsert" y contar el número
    de filas en cada grupo:
        day_of_week_group_count = total de filas en cada día de la semana de
                                   "dateInsert"
    Dividir el conteo por el número total de filas en el DataFrame:
        Para cada grupo del día de la semana de "dateInsert":
            day_of_week_size_ratio = day_of_week_group_count / total_filas

3. Calcular la proporción del tamaño por mes:
    Agrupar los datos por mes de "dateInsert" y contar el número de filas en
    cada grupo:
        month_group_count = total de filas en cada "month"
    Dividir el conteo por el número total de filas en el DataFrame:
        Para cada valor de mes de "dateInsert":
            month_size_ratio = month_group_count / total_filas

4. Calcular la proporción del tamaño por semana del año:
    Agrupar los datos por semana ISO del año de "dateInsert" y contar el
    número de filas en cada grupo:
        week_of_year_group_count = total de filas en cada "week_of_year"
    Dividir el conteo por el número total de filas en el DataFrame:
        Para cada valor de semana ISO del año de "dateInsert":
            week_of_year_size_ratio = week_of_year_group_count / total_filas

```

Listing 7.5: cálculo de proporciones del tamaño de las transacciones en función de diferentes períodos temporales.

3. Añadir comportamiento transaccional categorizado

Con el objetivo de enriquecer la variabilidad de los datos mediante información transaccional categorizada, se han identificado las siguientes categorías clave:

- **crypto**, relacionada con mensajes a los que se les aplican módulos criptográficos de seguridad.
- **schema**, que define el tipo de esquema utilizado.
- **type**, correspondiente al tipo de mensaje.

- `size_cat`, que clasifica los tamaños en categorías como pequeño, mediano y grande.
- `envSource_envDestination_transaction`, que representa las tuplas de interlocutores involucrados en la transacción.

A partir de la agrupación según las categorías de las variables, obtenemos en el listing 7.6:

- Promedio del tamaño por categoría (`<category>_size_ratio`).
- Z-score del tamaño agrupado por estas categorías.
- Desviación estándar del tiempo entre mensajes (`TSLM_std_per_<category>`).
- Frecuencia de transacciones por categoría (`transaction_frequency_<category>`).
- Proporción acumulativa del número de mensajes (`cumulative_<category>_ratio`).

```

1. Definir las columnas categóricas:
    columnas_categoricas = ["crypto", "schema", "type", "size_cat",
                            "envSource_envDestination_transaction2"]
2. Para cada columna en columnas_categoricas:
    a. Calcular la proporción por categoría:
        Agrupar los datos por la columna actual y calcular el promedio de
        "size" para cada categoría:
        size_ratio_por_categoria = promedio("size") para cada categoría en
        la columna actual
        Asignar el valor calculado a cada fila correspondiente.
    b. Calcular la desviación estándar del tiempo entre transacciones
        ("TSLM") por categoría:
        Agrupar los datos por la columna actual y calcular la desviación
        estándar de "TSLM":
        TSLM_std_por_categoria = desviación estándar("TSLM") para cada
        categoría en la columna actual
        Asignar el valor calculado a cada fila correspondiente.
    c. Calcular la frecuencia de transacciones por categoría:
        Agrupar los datos por la columna actual y calcular el tiempo promedio
        entre transacciones en "dateInsert":
        tiempo_promedio = diferencia promedio entre valores consecutivos en
        "dateInsert" en segundos
        Calcular la frecuencia como el inverso del tiempo promedio:
        frecuencia_transacciones = 1 / tiempo_promedio
        Asignar el valor calculado a cada fila correspondiente.
    d. Calcular la proporción acumulativa por categoría:
        Agrupar los datos por la columna actual y contar el número acumulativo
        de filas en cada categoría:
        conteo_acumulativo_por_categoria = posición de la fila dentro del
        grupo actual
        Dividir el conteo acumulativo por el total de filas en el DataFrame:
        proporción_acumulativa = conteo_acumulativo_por_categoria /
        total_filas
        Asignar el valor calculado a cada fila correspondiente.

```

Listing 7.6: agrupación de variables categóricas, cálculo de proporciones, desviaciones estándar, frecuencias de transacciones y proporciones acumulativas por categoría.

Por otro lado, en el listing 7.7 generamos:

- Tiempo entre transmisiones (`TSLM_<category>_<specific>`).
- Z-score del tiempo entre transacciones (`TSLM_<category>_<specific>_zscore`).

- Media y desviación estándar del tamaño (`sum_size_<category>_<specific>`, etc.).
- Z-score de la densidad transaccional (`transaction_density_<category>_<specific>_zscore`).
- Media ponderada del tamaño (`weighted_mean_size_<category>_<specific>`), ajustada por el tiempo entre mensajes.
- Variación porcentual del tamaño (`percent_change_size_<category>_<specific>` y su z-score).
- Z-score de frecuencia de transacciones (`<category>_<specific>_frequency_zscore`).

```
1. Para cada columna en columnas_categoricas:  
    a. Para cada categoría única en la columna actual:  
        i. Filtrar los datos correspondientes a la categoría actual:  
            datos_filtrados = Filtrar filas donde la columna actual coincide con  
                            la categoría.  
        ii. Calcular el tiempo entre transacciones (TSLM) en segundos:  
            TSLM = Diferencia de tiempo consecutiva en "dateInsert" para los  
                  datos filtrados.  
        iii. Asignar TSLM a las filas correspondientes y reemplazar valores  
             NaN por 0.  
        iv. Calcular y normalizar TSLM usando Z-Score:  
            TSLM_normalizado = (TSLM - promedio(TSLM)) /  
                               desviación_estándar(TSLM).  
        v. Calcular el tamaño promedio y desviación estándar para la categoría:  
            promedio_tamaño = promedio("size") en los datos filtrados.  
            desviación_tamaño = desviación_estándar("size") en los datos  
                               filtrados.  
        vi. Calcular y normalizar la densidad de transacciones usando Z-Score:  
            densidad_transacciones = 1 / promedio(TSLM).  
            densidad_normalizada = (densidad_transacciones -  
                                     promedio(global_densidad)) /  
                                     desviación_estándar(global_densidad).  
        vii. Calcular la proporción acumulativa del tamaño:  
            proporción_acumulativa = suma_acumulativa("size") /  
                                   suma_total("size") para los datos filtrados.  
        viii. Calcular el promedio ponderado por tiempo del tamaño:  
            diferencias_tiempo = Diferencia de tiempo consecutiva en  
                                  "dateInsert" en segundos.  
            promedio_ponderado = (suma("size" × diferencias_tiempo)) /  
                                 suma(diferencias_tiempo).  
        ix. Normalizar el promedio ponderado usando Z-Score:  
            promedio_ponderado_normalizado = (promedio_ponderado -  
                                              promedio(global_size)) / desviación_estándar(global_size).  
        x. Calcular la variación porcentual promedio del tamaño:  
            variación_porcentual = promedio(cambio_porcentual("size")) en los  
                                  datos filtrados.  
        xi. Normalizar la variación porcentual usando Z-Score:  
            variación_normalizada = (variación_porcentual -  
                                      promedio(global_cambio_porcentual)) /  
                                      desviación_estándar(global_cambio_porcentual).  
        xii. Calcular la tasa de repetición de la categoría:  
            tiempo_total = Tiempo total entre el primer y último valor de  
                           "dateInsert" en segundos.  
            frecuencia = Número de filas en los datos filtrados / tiempo_total.  
        xiii. Normalizar la frecuencia usando Z-Score:  
            frecuencia_normalizada = (frecuencia - promedio(global_densidad))  
                                      / desviación_estándar(global_densidad).  
2. Asignar todos los valores calculados a las columnas correspondientes en  
   el DataFrame original.
```

Listing 7.7: Agrupación mediante variables categóricas, cálculo y normalización de métricas como el tiempo entre transacciones, tamaño promedio, densidad de transacciones, proporciones acumulativas, promedio ponderado, variación porcentual y tasa de repetición por categoría.

4. Agrupación horaria

Con el objetivo de capturar variaciones horarias de las métricas previamente calculadas, regularizar los intervalos en la serie temporal y revelar patrones estacionales, se agrupan los datos por hora y se agrega la suma o media de las métricas para ese periodo según corresponda, este proceso se muestra en el listing 7.8.

1. Inicializar una lista de claves para agregaciones por suma:
`claves_suma = ["size", "intSize", "TSLM_crypto_SMIME", "TSLM_crypto_NO_CRYPTO", "TSLM_crypto_SEDI", ...].`
2. Inicializar una lista de claves para agregaciones por promedio:
`claves_promedio = ["size_deviation_from_mean", "size_z_score", "hour_size_ratio", ...].`
3. Crear un diccionario vacío para almacenar las reglas de agregación:
`diccionario_agregación = {}.`
4. Para cada clave en claves_suma:
 - a. Añadir una regla de agregación "suma" al diccionario:
`diccionario_agregación[clave] = ["sum"].`
5. Para cada clave en claves_promedio:
 - a. Añadir una regla de agregación "promedio" al diccionario:
`diccionario_agregación[clave] = ["mean"].`
6. Realizar la agregación por hora en el DataFrame:
 - Agrupar los datos usando el campo "dateInsert" con frecuencia de 1 hora.
 - Aplicar las reglas de agregación definidas en diccionario_agregación.
7. Renombrar las columnas del DataFrame resultante para mayor claridad:
 - Para cada clave y tipo de agregación en diccionario_agregación:
 - a. Crear un nuevo nombre en el formato "{clave}_{agregación}".
 - Asignar los nuevos nombres de columnas a df_final_hour.
8. Fin del proceso.

Listing 7.8: agrupación horaria, mediante la media o la suma, de las métricas calculadas.

Asimismo, en el listing 7.9 generamos el número total de mensajes transaccionados por hora de forma global y para cada categoría específica definida previamente.

1. Definir la lista de columnas categóricas:
`columnas_categóricas = ["crypto", "envSource_envDestination_transaction2", "schema", "type", "size_cat"].`
2. Para cada columna en columnas_categóricas:
 - a. Para cada categoría en los valores únicos de la columna actual:
 - i. Generar un nombre de columna para la cantidad de mensajes:
`nombre_columna = "msg_num_" + columna + "_" + categoría.`
 - ii. Calcular la cantidad de mensajes por hora para la categoría actual:
 - Filtrar las filas donde columna == categoría.
 - Agrupar por la columna "dateInsert" con frecuencia de 1 hora.
 - Contar el número de mensajes.
 - Rellenar los valores faltantes con 0.
 - iii. Agregar el resultado al DataFrame DF con el nombre de columna calculado.
 - iv. Si la columna actual no es "size_cat":
 - Generar un nombre de columna para la suma del tamaño:
`nombre_columna2 = "sum_size_" + columna + "_" + categoría.`
 - Calcular la suma de "size" por hora para la categoría actual:
 - Filtrar las filas donde columna == categoría.
 - Agrupar por "dateInsert" con frecuencia de 1 hora.
 - Sumar los valores de "size".

```

    - Rellenar los valores faltantes con 0.
    - Agregar el resultado al DataFrame DF con el nombre de columna
      calculado.

3. Calcular columnas globales:
df_global = Agrupar por "dateInsert" con frecuencia de 1 hora y contar el
            número total de mensajes.
Renombrar la columna resultante como "msg_num".
4. Fin del proceso.

```

Listing 7.9: cálculo del número de mensajes agrupando según periodo horario y variables categóricas.

5. Tasas relacionadas con la actividad transaccional

Con el objetivo de analizar las tasas de envío y espera para identificar patrones y anomalías, en el listing 7.10 se calculan dos métricas clave: la tasa de envío de mensajes por categoría, obtenida dividiendo el número total de mensajes intercambiados entre el tiempo total transcurrido desde el último mensaje, y la tasa de espera entre transmisiones, calculada dividiendo el tiempo total transcurrido entre la tasa de mensajes por hora.

```

1. Calcular tasas globales:
a. Message_Rate = msg_num / (TSLM_sum / 3600).
b. Message_Waiting_Rate = TSLM_sum / (msg_num / 3600).

2. Para cada columna en columnas_categóricas:
a. Para cada categoría en los valores únicos de la columna:
   i. Generar el nombre de la columna de número de mensajes para la categoría
      actual:
      msg_num_col = "msg_num_" + columna + "_" + categoría.
   ii. Generar el nombre de la columna de TSLM para la categoría actual:
      tsrm_sum_col = "TSLM_" + columna + "_" + categoría + "_sum".
   iii. Crear una nueva columna de tasa de mensajes por categoría:
      "Message_" + columna + "_Rate_" + categoría = msg_num_col / (
      tsrm_sum_col / 3600).
   iv. Crear una nueva columna de tasa de espera de mensajes por categoría:
      "Message_" + columna + "_Waiting_Rate_" + categoría = tsrm_sum_col /
      (msg_num_col / 3600).

3. Fin del proceso.

```

Listing 7.10: cálculo de tasas de actividad y densidad transaccional.

6. Lags temporales

Buscando capturar dependencias temporales y patrones recurrentes, se estudia la inclusión de retardos temporales significativos en las variables agregadas, que miden la relación entre una serie temporal y una versión rezagada de esta durante períodos sucesivos [41]. Para ello, se va a utilizar la métrica de autocorrelación, que mide la relación entre un periodo de la serie y su versión rezagada para distintos intervalos, ayudando a detectar patrones como estacionalidad y tendencias. En un gráfico de ACF, la altura de las barras representa el coeficiente de correlación en cada rezago, lo que facilita la identificación de dependencias significativas.[41].

La siguiente gráfica (figura 7.1) muestra la correlación de los rezagos para las variables size y msg_num. Las líneas verticales rojas delimitan intervalos de 24 horas, facilitando una visualización más clara y ordenada. Una alta correlación en el ACF indica que los valores pasados influyen fuertemente en los futuros, reflejando relaciones entre transacciones a lo largo del tiempo. Si bien se observa un patrón diario con picos a 24 rezagos, los mayores valores de correlación se presentan en los rezagos semanales, especialmente en múltiplos de 168 lags, como 336, 504, 672 y 840,

destacando la importancia del rezago 168 como un factor clave en la dinámica del tráfico.

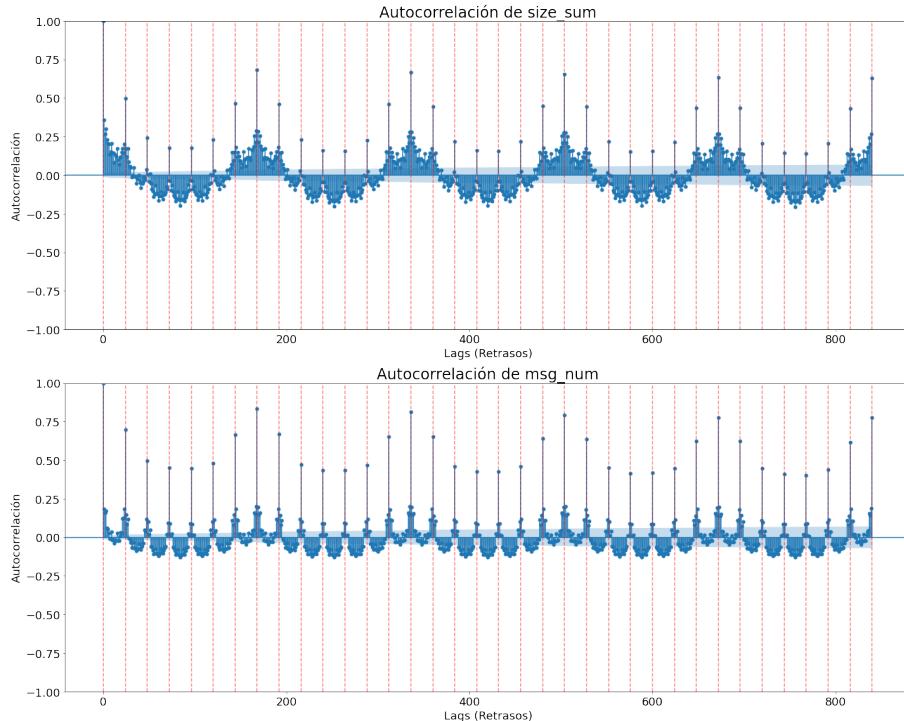


Figura 7.1: análisis de la correlación por rezagos.

Una vez confirmado que existen dependencias temporales, se busca identificar los rezagos temporales más relevantes a considerar. La combinación de métricas de correlación permite reconocer estructuras subyacentes en los datos, aspecto clave para la modelización de series temporales y la toma de decisiones basadas en patrones históricos [41].

En el listing 7.11, los rezagos clave para cada variable del conjunto se almacenan en `resultados_lags` para su posterior uso. Estos se identifican mediante tres métricas principales: Autocorrelación (ACF), correlación parcial (PACF), que mide la correlación entre observaciones en distintos momentos del tiempo, eliminando la influencia de rezagos intermedios para aislar relaciones directas [41] y por último, correlación cruzada (CCF), empleada para analizar la relación con las variables de tamaño y número de mensajes, ya que son las características originales más relevantes para diferenciar transacciones.

```
Función calculate_top_lags(data, target_vars, columns_to_process, max_lag=336,
    top_n=5, min_gap=5, ccf_threshold=0.2):
    Inicializar lista results
    Inicializar diccionario lag_dict como vacío # Para almacenar los lags de
        # cada predictor y objetivo

    # Variables predictoras
    predictors = Filtrar columnas en columns_to_process que no están en
        target_vars

    # Función para filtrar lags redundantes, priorizando aquellos con alta
    # correlación y grandes diferencias
    Función filter_redundant_lags(lags, correlations):
        Inicializar lista selected_lags vacía # Para almacenar sólo los lags
        Inicializar lista selected_lags_with_corr vacía # Para almacenar lags
            con sus correlaciones
        Para cada par de lag y correlación en lags y correlations:
```

```

    Si todos los lags en selected_lags_with_corr están a una distancia
    mayor a min_gap:
        Añadir (lag, corr) a selected_lags_with_corr
        Añadir lag a selected_lags
    Si el número de lags seleccionados es mayor o igual a top_n:
        Salir del bucle
    Retornar selected_lags

# Calcular CCF para cada combinación de predictor y objetivo
Para cada target en target_vars:
    Para cada predictor en predictors:
        Calcular CCF entre data[target] y data[predictor] hasta max_lag
        Ordenar los lags por la correlación absoluta del CCF
        Filtrar lags redundantes usando filter_redundant_lags
        Seleccionar los lags significativos basados en el umbral ccf_threshold
        Si hay lags significativos:
            Si predictor no está en lag_dict:
                Inicializar lista vacía en lag_dict[predictor]
                Añadir los lags significativos a lag_dict[predictor]

# Calcular lags de ACF y PACF para las variables objetivo
Iniciar diccionario target_lags
Para cada target en target_vars:
    Calcular ACF y PACF hasta max_lag
    Seleccionar los lags más relevantes de ACF y PACF utilizando
        filter_redundant_lags
    Combinar los lags de ACF y PACF, eliminando duplicados
    Ordenar los lags combinados por la correlación de ACF y seleccionar los
        top_n
    Guardar los lags seleccionados en target_lags[target]

# Combinar los lags de cada predictor, priorizando aquellos con mayor
correlación
Iniciar lista combined_results
Para cada predictor y lags en lag_dict:
    Crear un DataFrame lag_counts con lags y correlaciones
    Agrupar lag_counts por "lag" y calcular la media de las correlaciones
    Calcular el conteo de cada lag
    Ordenar top_lags por correlación y conteo de lags en orden descendente
    Añadir los resultados de top_lags a combined_results

# Añadir los lags relevantes para las variables objetivo
Para cada target y lags en target_lags:
    Para cada lag en lags:
        Añadir el lag a combined_results con predictor como target

    Retornar un DataFrame de combined_results

```

Listing 7.11: identificación de rezagos temporales más significativos mediante ACF, PACF y CCF.

En el listing 7.12, se calculan variables agregadas a partir de los rezagos previamente determinados para capturar dependencias temporales. Mediante el uso de ventanas móviles, se reduce el ruido en los datos y se resaltan mejor las tendencias subyacentes, lo que facilita la identificación de patrones relevantes. Además, la incorporación de retardos significativos optimiza la representación de la estructura temporal, mejorando la capacidad del modelo para aprender relaciones dinámicas en la serie temporal [47]. A continuación las variables agregadas a partir de retardos temporales:

- Lags: se crean columnas con valores desplazados en el tiempo de acuerdo con los lags más relevantes indicados en results_lags. <col>_lag_<lag>.

- Diferencias de lags: representan la diferencia entre el valor actual y el valor de un lag específico. (<col>_diff_lag_<lag>).
- Media móvil (<col>_rolling_mean_<lag>).
- Desviación estándar móvil: desviación estándar calculada en una ventana móvil <col>_rolling_std_<lag>.
- Pendiente móvil (Rolling Slope): pendiente de una regresión lineal ajustada dentro de una ventana móvil. <col>_rolling_slope_<lag>.
- Frecuencia de cambios significativos: número de cambios mayores a un umbral dentro de una ventana móvil. <col>_frequency_changes_<lag>.
- Desviación estándar acumulativa: desviación estándar calculada desde el inicio de la serie hasta un punto en el tiempo, con un mínimo de periodos de lag para evaluar la variabilidad acumulada. <col>_cumulative_std_<lag>.
- Interacciones entre las variables originales y variables cíclicas temporales: producto de las columnas originales con variables que representan ciclos temporales como hora, día de la semana, mes, etc. (<col>_x_hour_cos).

```

Función rolling_slope(series, window):
    Inicializar lista slopes vacía
    Para cada ventana de tamaño window en series:
        Calcular los coeficientes de la regresión lineal utilizando np.polyfit
        Almacenar la pendiente (coeficiente de la regresión) en slopes
    Retornar slopes

Función frequency_of_changes(series, window, threshold_percentile):
    Calcular el percentil especificado de la serie como umbral
    Para cada valor en la diferencia absoluta de la serie:
        Si el cambio absoluto es mayor que el umbral, marcarlo como cambio
        significativo
    Calcular la suma de cambios significativos en ventanas móviles de tamaño
        window
    Retornar la frecuencia de cambios

Función generate_features_with_lags(df, cols, results_lags):
    Crear una copia del DataFrame df en df_features

    # Crear lags basados en los resultados_lags
    Para cada columna col en cols:
        Obtener los lags relevantes de results_lags para col
        Para cada lag en los lags relevantes:
            Crear una nueva columna en df_features con los lags (shift) de col
            Rellenar los valores nulos con 0

    # Crear diferencias con lags basados en los lags más relevantes
    Para cada columna col en cols:
        Obtener los lags relevantes de results_lags para col
        Para cada lag en los lags relevantes:
            Crear una nueva columna con la diferencia de col y su lag
            Rellenar los valores nulos con 0

    # Crear medias móviles y otras características basadas en los lags
    Para cada columna col en cols:
        Obtener los lags relevantes de results_lags para col
        Para cada lag en los lags relevantes:
            Calcular la media móvil de col con ventana de tamaño lag
            Calcular la desviación estándar de col con ventana de tamaño lag
            Rellenar los valores nulos con 0

    # Calcular pendiente (rolling slope)

```

```

Si lag es mayor o igual a 2:
    Calcular la pendiente con la función rolling_slope y agregarla a
        df_features
    Rellenar los valores nulos con 0

# Calcular frecuencia de cambios significativos
Calcular la frecuencia de cambios significativos con la función
    frequency_of_changes y agregarla a df_features
Rellenar los valores nulos con 0

# Calcular desviación estándar acumulada
Calcular la desviación estándar acumulada con expanding y agregarla a
    df_features
Rellenar los valores nulos con 0

# Crear interacciones con variables cíclicas
Definir las variables cíclicas como hour_cos, hour_sin, day_of_week_cos,
    day_of_week_sin, month_sin, month_cos, week_of_year_sin, week_of_year_cos
Para cada columna col en cols:
    Para cada variable cíclica en las variables cíclicas:
        Crear una nueva columna como el producto de col y la variable cíclica
        Almacenar el resultado en df_features

Retornar df_features

```

Listing 7.12: cálculo de variables mediante ventanas móviles y rezagos temporales, basados en los rezagos más significativos previamente identificados en el análisis de autocorrelación.

Tras incorporar las nuevas características y agrupar los datos por períodos horarios, se obtiene una serie temporal regular de 25.840 horas y 6.437 variables. Esta incluye tanto variables agregadas como aquellas basadas en rezagos, diseñadas para capturar patrones transaccionales a lo largo de distintos períodos de tiempo.

7.2 Reducción de dimensionalidad

La agregación de variables y características, ha generado un conjunto de datos con alta dimensionalidad, lo que puede afectar la eficiencia y el rendimiento del entrenamiento de los modelos. Para abordar este problema, se emplea Análisis de Componentes Principales (**PCA**), que permite reducir la cantidad de características preservando la información más relevante [42].

Para comenzar el proceso, las características se han escalado con *StandardScaler*, ajustando su media a cero y su desviación estándar a uno. Este paso es fundamental para asegurar que todas las variables contribuyan equitativamente, evitando que aquellas con valores más amplios dominen sobre las de menor escala [32].

Para mejorar la efectividad e interpretabilidad del **PCA** y garantizar que las componentes principales capturen información diversa, en el listing 7.13, se eliminarán redundancias filtrando variables altamente correlacionadas, con un umbral de 0,9. Esto evita que la representación de la varianza esté sesgada por relaciones redundantes y permite que el **PCA** revele patrones no evidentes, facilitando una interpretación más clara [32].

- 1.Calcular la matriz de correlación de los datos filtrados (filtered_data_dfn):


```
correlation_matrix = correlación de filtered_data_dfn
```
- 2.Establecer el umbral de correlación (correlation_threshold):


```
correlation_threshold = 0.9
```
- 3.Obtener los pares de correlación alta:


```
Desapilar la matriz de correlación (transformar la matriz en una lista de pares)
      Ordenar los pares en orden descendente según su valor de correlación
```
- 4.Filtrar los pares con correlación mayor que el umbral:

```

high_corr_pairs = pares con correlación > 0.9
Excluir pares con correlación igual a 1.0 (correlación perfecta consigo mismo)
5.Inicializar un conjunto vacío (to_drop) para almacenar las columnas a eliminar
6.Recorrer los pares de columnas correlacionadas y decidir cuál eliminar:
    Para cada par (col1, col2) en high_corr_pairs:
        Si col1 y col2 no están en to_drop:
            Agregar col2 a to_drop (eliminar la segunda columna del par)
7.Eliminar las columnas identificadas en to_drop de los datos filtrados
(filtered_data_dfn):
filtered_data_dfn = filtered_data_dfn sin las columnas de to_drop

```

Listing 7.13: filtrado de variables redundantes mediante la identificación de alta correlación mutua.

Para maximizar la varianza y reducir las dimensiones, es fundamental determinar el número óptimo de componentes a seleccionar. Para ello, se analiza la varianza acumulada y el método del codo en la figura 6.19, donde se observa que, a partir de la décima componente, la ganancia de varianza capturada se reduce significativamente, mientras que en las primeras siete el incremento es más notable.

Por otro lado, en el gráfico del método del codo mostrado en la figura 7.2, la línea azul representa la inercia, que ilustra cómo disminuye la dispersión de los datos a medida que aumenta el número de componentes principales. Al principio, la inercia disminuye de manera pronunciada, lo que indica que los primeros componentes explican gran parte de la variabilidad de los datos. Sin embargo, a medida que se añaden más componentes, el descenso se vuelve más gradual, sugiriendo que los componentes adicionales aportan cada vez menos. Por su parte, la línea verde refleja la varianza explicada acumulada, que indica cuánta variabilidad total de los datos es capturada por los componentes. Aunque la varianza explicada aumenta con cada componente, la tasa de incremento disminuye con el tiempo, lo que sugiere que, después de cierto número de componentes, la mejora en la explicación de la varianza es menos significativa. En el gráfico se observa una línea que indica que el valor óptimo de componentes es $k=10$, siendo este el número de componentes donde la combinación de la inercia y la varianza acumulada alcanza un punto de equilibrio.

En línea con el enfoque del **PCA**, las componentes se ordenan según su importancia, de modo que las más relevantes explican la mayor parte de la variabilidad en los datos [32].

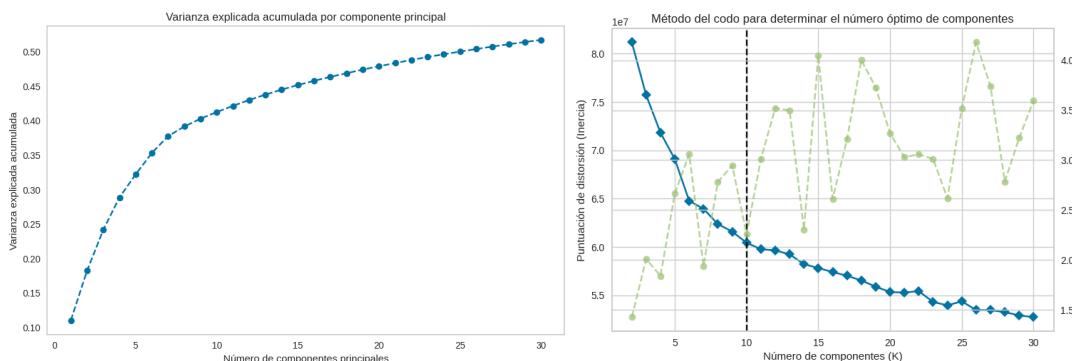


Figura 7.2: varianza explicada y método del codo para PCA.

La elección de las primeras 10 componentes principales explicaría aproximadamente el 40 % de la varianza. A pesar de que esta selección no capture su plenitud, dado el elevado número de características originales, se espera que el **PCA** conserve la información más relevante para diferenciar los patrones transaccionales. Esto no solo reduce la complejidad de los datos y por consiguiente la interpretación de los resultados sino que también

mejora la escalabilidad del análisis en datos de alta dimensión al reducir la cantidad de variables de entrada al modelo de clustering [42].

Observamos en la figura 7.3 las interacciones entre componentes mediante un ráfico de pares (pairplot). A primera vista vemos que a medida que nos alejamos de **PC1**, se aprecia una mayor dispersión de los puntos, indicando una mayor presencia de ruido y una separación menos efectiva.

Por otro lado, las interacciones entre componentes denotan agrupaciones visibles, sugiriendo que las componentes principales son capaces de diferenciar clases a partir de ellas. Esto puede verse de forma clara en la interacción entre **PC1** y **PC2**, **PC2** y **PC4/ PC5**, donde pueden discernirse tres grupos separados por dichas componentes. Podemos concluir que **PCA** ha logrado una reducción efectiva de la dimensionalidad, conservando la información relevante para distinguir los grupos en el espacio reducido.

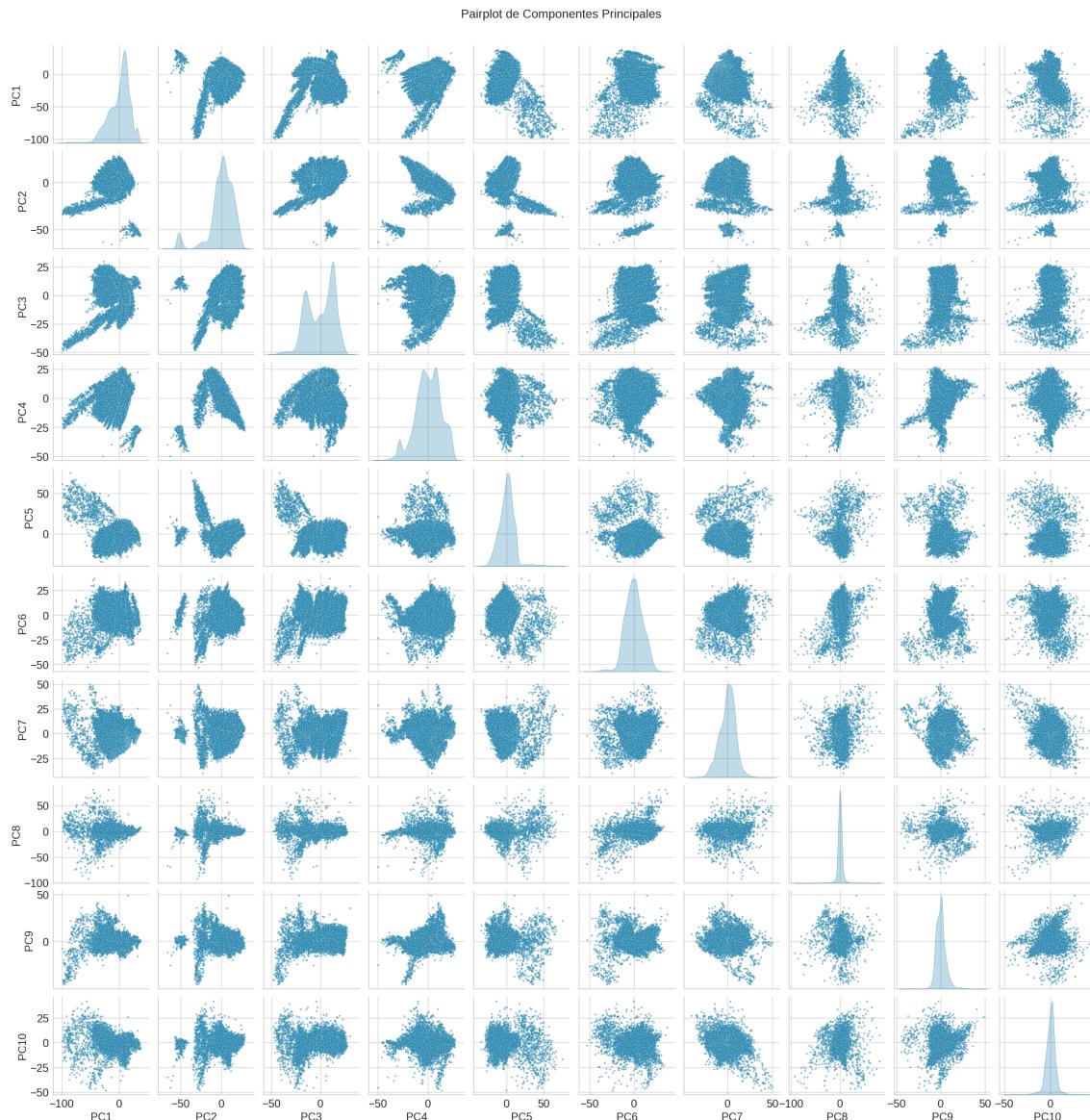


Figura 7.3: gráfico de pares de las componentes extraídas.

Una vez reducido el conjunto a utilizar a partir de los loadings, estudiaremos para las componentes principales obtenidas qué características y de qué forma contribuyen a la explicación de la varianza:

- **PC1: primera componente principal**

- **Contribuciones positivas**

Contribuciones de variables relacionadas con la variable cíclica horarias coseno, como **Msg_Rate_x_hour_cos**, **hour_size_ratio_mean_x_hour_cos** y **hour_intSize_ratio_mean_x_hour_cos** estas son positivas durante la primera mitad de la madrugada y primera mitad de la tarde (00:00 - 06:00 /12:00 - 18:00), lo refleja un pico tanto del tránsito como del tamaño de los mensajes durante estos periodos, a la par que una depresión posterior.

Por otro lado, también se destaca la contribución positiva de la tasa de cambio en el tamaño de los mensajes **size_rate_of_change_sum** contribuyendo a capturar cambios en el tamaño de los contenidos de los mensajes.

De forma general, las variables que contribuyen positivamente a la primera componente, reflejan un comportamiento transaccional concentrado durante primera mitad de la madrugada y primera mitad de la tarde, asociado a altos volúmenes y variabilidad del tamaño de mensajes. Esto podría ser relevante para diseñar estrategias específicas o para optimizar recursos durante este periodo.

- **Contribuciones negativas**

En cuanto a las variables que contribuyen negativamente a esta componente, podemos observar variables que miden el número de mensajes de tamaños medios, tanto altos como bajos. Como son **msg_num_Medium_High** y sus rezagos (**168, 336, 504** y **672**) y **msg_num_Medium_Low**. También se encuentran presentes variables que miden cambio del tamaño a través de las semanas del año como, como lo es **size_week_of_year_change_sum**.

Además, se encuentra presente la variable **cum_size_ratio_121_50_mean**, sus medias móviles y rezagos temporales **rolling_mean_1**, **lag_336**, **lag_672** correspondientes al ratio acumulado del contenido de los mensajes para algunos interlocutores.

Finalmente, se denota la presencia de variables relacionadas con el protocolo de seguridad, en específico aquellas sin configuración de protocolo **NO CRYPTO**. Estas contribuyen negativamente, tanto en la tasa transaccional, debido a la presencia de **Msg_crypto_Rate_NO CRYPTO** y sus rezagos (**lag_168**, **lag_336**, **lag_504**), como también en el ratio del tamaño, mediante **hour_size_ratio_by_crypto_mean** y su rezago con el periodo 672.

Se concluye que, durante los picos de actividad de los periodos señalados, los tamaños de mensaje en el rango medio, tanto altos como bajos son poco predominantes, siendo más comunes en horarios o días de menor actividad. Además, cambios significativos en el tamaño de los mensajes a lo largo de las semanas tienen menor relevancia durante estos periodos de alta actividad, donde también ciertos interlocutores o flujos específicos participan menos. En contraste, los intercambios con protocolos de seguridad tienden a dominar durante los picos matinales, lo que sugiere que las operaciones más críticas o sensibles suelen concentrarse en estos periodos.

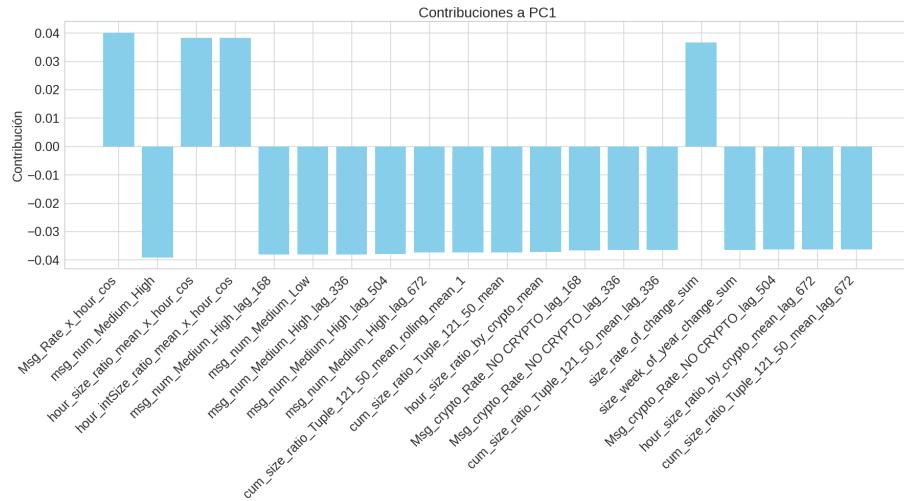


Figura 7.4: contribuciones de variables a la primera componente principal.

En resumen, valores altos de **PC1** se asocian con horas durante la primera mitad de la madrugada y primera mitad de la tarde caracterizadas por una mayor transmisión de mensajes, existiendo en estos variabilidad en el tamaño. También se asume patrones horarios con un elevado número de mensajes con protocolos de seguridad activos y pocas transmisiones del interlocutor **50**. Por otro lado, valores bajos de **PC1** corresponden a horas fuera del rango anterior con una menor actividad transaccional, mayor presencia de tamaños medios, y más variabilidad en el tamaño de los mensajes comparando semanalmente. Además, en los períodos con valor bajo para **PC1** se espera un aumento del número de intercambios sin protocolos de seguridad.

■ PC2: segunda componente principal

- **Contribuciones positivas**

En primer lugar, la presencia de `type_size_ratio_mean`, que mide el ratio de tamaño según la dirección de los mensajes, junto con la `std_size_type_S_mean`, que representa la desviación estándar promedio del tamaño de los mensajes de salida, indican que esta componente captura la variabilidad en el tamaño de los mensajes dependiendo de su dirección. Esto refleja diferencias operativas asociadas al flujo de datos entrantes y salientes en el sistema.

Además, variables como `mean_size_crypto_SEDI_mean`, que mide el tamaño promedio según el protocolo de seguridad **SEDI**, y `Wgt_mean_size_High_mean`, que se relaciona con los tamaños elevados de mensajes, sugieren una fuerte relación entre tamaños más altos y protocolos de seguridad específicos. Esto implica que esta componente captura aspectos relacionados con la seguridad de las transacciones, destacando cómo el uso del protocolo SEDI tiende a asociarse con mensajes de mayor tamaño.

Por otro lado, se destaca que la varianza explicada no solo está influenciada por características generales de los mensajes, sino también por dinámicas específicas de las relaciones entre entidades. Demostrado por las contribuciones de variables como `Wgt_mean_size_envSource_envDestination_*`, que resaltan patrones operacionales recurrentes entre ciertos interlocutores como los interlocutores **175**, **258** y **329**. También, la variabilidad en el tamaño para ciertos flujos, ya que la aparición de `std_size_121_50_mean` y `std_intSize_121_175_mean` destacan la variabilidad en el tamaño de los contenidos del flujo con el interlocutor **50** y la variabilidad de las interacciones con el interlocutor **175**.

Estas contribuciones reflejan cómo la segunda componente principal captura tanto la variabilidad en los tamaños de los mensajes según su dirección, protocolos de seguridad y categorías de tamaño, como los perfiles de actividad específicos de ciertas entidades.

- Contribuciones negativas

Se destacan para esta componente la contribución negativa de variables que reflejan patrones relacionados con el tamaño de los contenidos, las interacciones, y las dinámicas de ciertos interlocutores. Estas contribuciones negativas reflejan comportamientos opuestos a los capturados por las variables con contribuciones positivas, aportando una visión complementaria sobre las características de los datos.

Asimismo, se observa una contribución negativa asociada al tamaño medio ponderado de los contenidos y las interacciones en mensajes relacionados con respuestas a pedidos (**ORDRSP**) y facturas (**INVOIC**). Esto se evidencia a partir de las contribuciones de variables como

Wgt_mean_size_ORDRSP_zscore_mean y
Wgt_mean_intSize_INVOIC_zscore_mean.

También se encuentran presentes variables relacionadas con el tamaño de los mensajes de tamaño medio-altos y el tamaño de interacciones específicas de ciertos interlocutores, como el **102**, debido a las variables

Wgt_mean_size_Medium_High_zscore_mean y
Wgt_mean_size_102_121_zscore_mean. Además, a través de la variable
pct_change_size_*, se destacan comportamientos diferenciales respecto a la
tasa de cambio del tamaño en mensajes para interlocutores específicos como
el 399 y el 388.

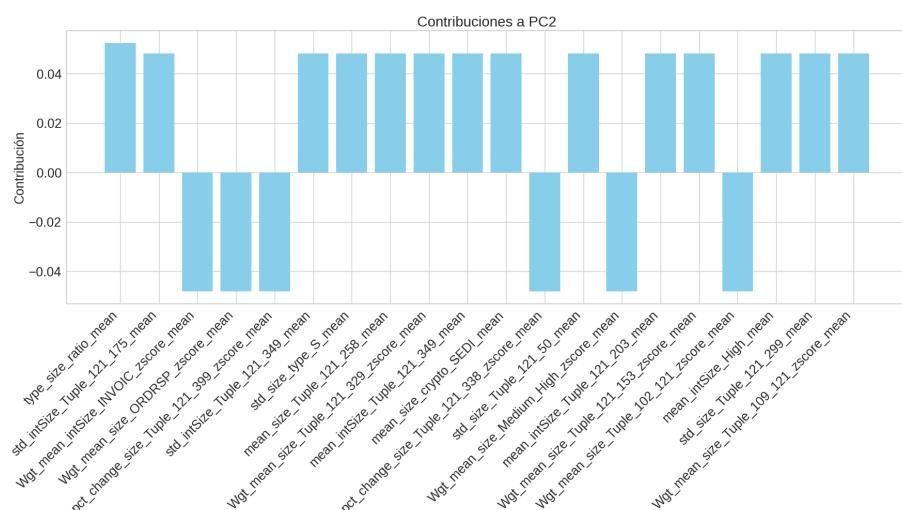


Figura 7.5: contribuciones de variables a la segunda componente principal.

En resumen, periodos de interacciones con tamaños más elevados de **PC2** representan patrones para los cuales los mensajes probablemente tienen mayor tamaño, predominando protocolos de seguridad como SEDI y patrones consistentes entre entidades como los interlocutores **175, 258, 329** de forma recurrente. También se detecta heterogeneidad del tamaño de mensajes dentro de estos flujos, pudiendo existir cambios bruscos en esta característica. Por otro lado, los valores más bajos para **PC2** se asocian a periodos de tránsito horarios caracterizados por mensajes de tipo respuestas a pedidos y facturas y con un tamaño más modesto, pero con un tránsito más regular y sin picos de actividad y más vinculados con otros protocolos

de seguridad como **SMIME/ NO CRYPTO**. También, interacciones con interlocutores como **102, 399** y **388** presentan mayores tasas de variación en el tamaño.

- **PC3: tercera componente principal**

- **Contribuciones positivas**

Las variables con contribución positiva para esta componente refleja un gran volumen de tránsito y una alta variabilidad en el tamaños de los mensajes, particularmente de periodos horarios comprendidos entre la primera mitad de la mañana y la última mitad de la tarde (06:00 - 12:00 /18:00 - 24:00), según lo indicado por las variables cíclicas (*hour_sin*). Estas contribuciones están asociadas a interlocutores específicos como **68, 313** y **159**, y están influenciadas por patrones que destacan picos de actividad operativa.

Variables como *Wgt_mean_size_*_hour_sin* y *pct_change_size_*_hour_sin* indican que los valores altos de la componente se asocian a periodos transaccionales donde existan fluctuaciones significativas en el tamaño y destaque los mensajes grandes, lo que sugiere un comportamiento dinámico y de alta intensidad entre los interlocutores mencionados.

En resumen, los valores altos en esta componente indican dinámicas de tránsito con alta cadencia, grandes volúmenes de contenido, y picos de actividad concentrados en periodos horarios específicos.

- **Contribuciones negativas**

Las contribuciones negativas representan un comportamiento opuesto, con transacciones de menor volumen y menor variabilidad en el tamaño del contenido. Estas están asociadas a interlocutores como **122, 202, 203, 291** y **390**, también se observan durante los mismos intervalos horarios, que son entre la primera mitad de la mañana y la última mitad de la tarde.

Variables como *mean_size_*_hour_sin* y *mean_intSize_Medium_High_mean_x_hour_sin* señalan que las interacciones en esta categoría tienen tamaños más pequeños o medio-altos, pero con menor volatilidad. Este comportamiento sugiere flujos más regulares y menos dinámicos en comparación con las contribuciones positivas.

Se puede concluir que esta componente refleja intercambios con menor volumen, menor variabilidad, y patrones más estables y predecibles.

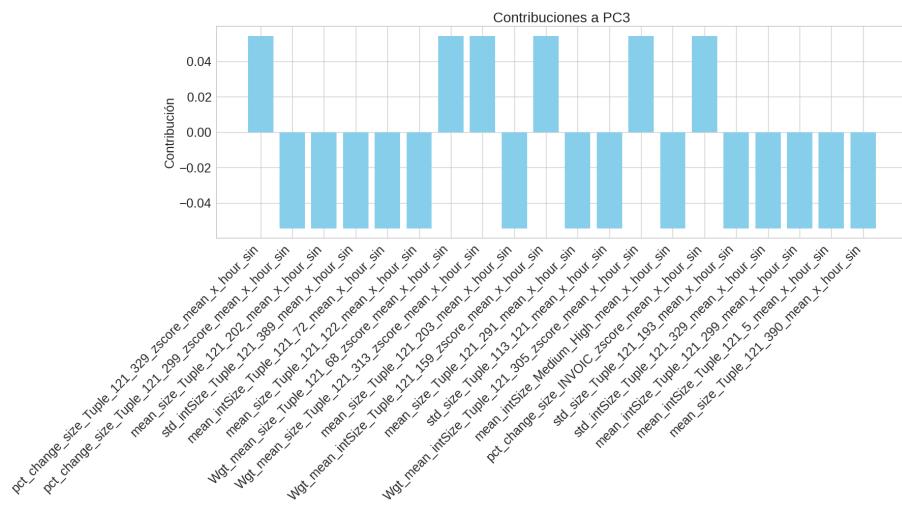


Figura 7.6: contribuciones de variables a la tercera componente principal.

En resumen, valores altos para la tercera componente principal, indican dinámicas de alto volumen y alta variabilidad en el tamaño del contenido, particularmente

entre los interlocutores **68, 313 y 159**, con picos de actividad en la mañana y tarde. Esto refleja intensidad operativa y volatilidad en flujos clave. Por otro lado, valores bajos representan transacciones de menor volumen y menor variabilidad, relacionadas con flujos más estables y regulares, destacando la actividad de interlocutores como **122, 202, 203, 291 y 390** durante los mismos periodos horarios.

A continuación, se ha realizado un análisis general de las variables que contribuyen a explicar la varianza de las restantes componentes principales. Para un análisis más detallado por cada componente, puede consultarse el apartado 11.2 de los anexos. Las siguientes características y sus derivados explican la varianza para el tamaño y número de mensajes dentro de las dinámicas de interacciones horarias de nuestro cliente:

- **Patrones cílicos semanales:** representados por `day_of_the_week_sin`. Esta variable es positiva aproximadamente entre los días 1 y 3,5 (lunes a miércoles al mediodía) y negativa entre 3,5 y 7 (miércoles por la tarde a domingo), lo que refleja diferencias significativas entre los patrones del inicio y el final de la semana.
- **Relación entre periodos temporales:** rezagos específicos (**24, 168, 336, 504 y 672**, entre otros) que capturan recurrencias cílicas diarias, semanales y mensuales.
- **Tipos de mensajes:** contribuciones significativas de ciertos tipos de mensajes, como **GENERAL, ORDERS, ORDRSP y OSTRPT**.
- **Categorías de tamaño:** impacto significativo de las categorías de tamaño, como **Medium Low, Medium High y High**.
- **Otros factores:**
 - Interacciones clave entre interlocutores específicos y flujos transaccionales (entrantes y salientes).
 - Contribuciones de mensajes sin protocolo criptográfico (**NO_CRYPTO**).
 - Métricas acumulativas, variabilidad diaria y tendencias móviles (rolling slopes), que reflejan cambios promedio y dinámicas dentro de ventanas temporales específicas.

Una vez obtenida esta visión general sobre las componentes principales y cómo las variables de los datos contribuyen a ellas, vamos a estudiar su eficacia para agrupar los datos mediante los modelos de clustering propuestos.

7.3 Entrenamiento, evaluación y validación de modelos de segmentación

Esta sección se centra en el entrenamiento y la evaluación de los métodos seleccionados para la segmentación de grupos mediante los modelos **K-Means** y **HDBSCAN**, describiendo su proceso de ajuste, selección de hiperparámetros y optimización. Para evaluar la calidad de los clusters generados, se utilizarán tres métricas de validación, las cuales se describen más adelante.

- **Índice de Silueta** evalúa la calidad de los clusters comparando la cohesión interna con la separación entre clusters. Valores cercanos a 1 indican buenos agrupamientos, mientras que valores cercanos a -1 sugieren asignaciones incorrectas.
- **Índice de Davies-Bouldin** mide la compacidad interna y la separación entre clusters. Un valor bajo refleja clusters más compactos y mejor separados.

- *Índice de Calinski-Harabasz* analiza la relación entre la dispersión interna y entre clusters. Valores altos indican una mejor separación y mayor compacidad en la estructura de los clusters.

Para evaluar la calidad de un agrupamiento, es recomendable utilizar múltiples métricas, ya que cada una captura distintos aspectos de la estructura de los datos. El coeficiente de Silhouette tiende a subestimar la cantidad de clusters cuando hay una alta variabilidad, dificultando la diferenciación de actividades similares. El índice Davies–Bouldin favorece clusters con baja variabilidad interna y alta separación, pero es sensible al ruido y a los valores atípicos. Por otro lado, el índice Calinski–Harabasz equilibra la compacidad y la separación, lo que facilita la identificación de grupos bien definidos sin segmentación excesiva. Sin embargo, en algunos casos, especialmente cuando el número de clusters es muy pequeño o el de muestras es muy grande, puede sobreestimar la cantidad óptima de clusters [52]. La combinación de estas métricas proporciona una evaluación más robusta del clustering, compensando sus limitaciones individuales y ofreciendo una visión más equilibrada de la calidad de la agrupación.

7.3.1. HDBSCAN

Se ha implementado una búsqueda de hiperparámetros para el algoritmo de clustering **HDBSCAN** mediante Optimización Bayesiana, explorando distintas combinaciones dentro de los rangos definidos en el listing 7.14 para identificar las configuraciones más prometedoras. Los parámetros clave incluyen **min_cluster_size**, que define el tamaño mínimo para formar un clúster; **min_samples**, que establece la densidad requerida al determinar la cantidad mínima de puntos para considerar uno como núcleo; y **cluster_selection_epsilon**, que delimita las fronteras entre clústeres, permitiendo identificar subestructuras dentro de los grupos. Ajustar adecuadamente estos parámetros es fundamental para capturar con precisión la estructura de los datos y obtener agrupamientos coherentes.

```
search_space = [Integer(5, 1000, name="min_cluster_size"),
                Integer(1, 1000, name="min_samples"),
                Real(0.01, 1.0, name="cluster_selection_epsilon")]
```

Listing 7.14: configuración inicial del espacio de búsqueda de hiperparámetros para el ajuste del modelo HDBSCAN.

Como se muestra en el listing 7.15, se ha definido una función objetivo que combina las tres métricas estudiadas para evaluar la calidad de los clusters generados por el modelo. Esta función se ha utilizado para probar diversas combinaciones de hiperparámetros , con el objetivo de optimizar el rendimiento del modelo.

1. Se crea una instancia del modelo **HDBSCAN** utilizando los valores actuales de los hiperparámetros a evaluar. Tras ajustar el modelo al conjunto de datos df, se generan etiquetas para cada punto, donde aquellos no asignados a ningún clúster se clasifican como ruido, representados con el valor -1.
2. Si el modelo genera un único clúster, etiqueta todos los puntos como ruido (-1) o asigna más del 50 % de los puntos al ruido, se omite el cálculo de las métricas y se aplica una penalización. En estos casos, se asigna un puntaje altamente negativo para desincentivar dichas configuraciones durante el proceso de optimización.
3. Como se ha mencionado anteriormente, la calidad de los clusters se evaluará considerando tres métricas clave, cuya ponderación busca equilibrar compacidad, separación y estabilidad. El Silhouette Score (50 %) tiene el mayor peso, ya que mide simultáneamente la cohesión dentro de los clusters y la separación entre ellos.

El inverso del Davies-Bouldin Index (30 %) contribuye a la evaluación penalizando clusters con alta variabilidad interna y baja separación. Por último, el Calinski-Harabasz Index (20 %) complementa el análisis al medir la relación entre la dispersión interna y la externa, siendo normalizado mediante el logaritmo del tamaño del conjunto de datos para evitar sesgos en grandes volúmenes de datos. Como el optimizador busca minimizar la función objetivo, el puntaje resultante se multiplica por -1, generando así una única métrica global que refleja de manera equilibrada la calidad del agrupamiento.

4. Finalmente, se aplica la optimización bayesiana mediante `gp_minimize`, con el objetivo de minimizar la función objetivo y encontrar los valores óptimos de los hiperparámetros , maximizando así la calidad del clustering.

```

Función objetivo(min_cluster_size, min_samples, cluster_selection_epsilon):
    Crear un objeto clusterer de HDBSCAN con los parámetros proporcionados:
        min_cluster_size = min_cluster_size
        min_samples = min_samples
        cluster_selection_epsilon = cluster_selection_epsilon

    Ajustar el clusterer al DataFrame df y obtener las etiquetas (labels)

    # Penalización por ruido:
    Si el número de etiquetas únicas es 1 o el porcentaje de etiquetas -1 (ruido) es
        mayor al 50%:
        Retornar un valor grande (1e6) como penalización

    Calcular el índice de silueta (silhouette) con df y las etiquetas
    Calcular el índice Davies-Bouldin (davies_bouldin) con df y las etiquetas
    Calcular el índice Calinski-Harabasz (calinski_harabasz) con df y las etiquetas

    # Función objetivo combinada:
    Calcular un puntaje negativo basado en la siguiente fórmula:
        score = -(
            (silhouette * 0.5) +
            ((1 / (1 + davies_bouldin)) * 0.3) +
            (calinski_harabasz / log(len(df))) * 0.2
        )
    Retornar el score

```

Listing 7.15: búsqueda bayesiana para optimizar la combinación de hiperparámetros para el modelo HDBSCAN.

Tras identificar la combinación óptima de hiperparámetros : `cluster_selection_epsilon` en 0,2342, `min_cluster_size` en 54 y `min_samples` en 1000, se entrenó el modelo **HDBSCAN** y se evaluó su desempeño utilizando métricas de calidad de clustering.

El Silhouette Score de 0,4776 sugiere una cohesión y separación moderadas entre los clústeres. El índice Davies-Bouldin de 0,8864 indica una relación favorable entre la variabilidad interna y la separación entre grupos, lo que sugiere clusters bien definidos. Por su parte, el índice Calinski-Harabasz alcanza un valor de 5398,36, lo que refuerza la estructura del agrupamiento al evidenciar una buena separación relativa entre los clústeres. En conjunto, estas métricas reflejan un modelo con un desempeño aceptable y una segmentación estructurada, aunque con margen de mejora.

HDBSCAN construye un árbol de mínima expansión para organizar los datos en una jerarquía de clusters y filtra los menos estables según su densidad y duración en la jerarquía. El resultado es un árbol condensado que conserva únicamente las agrupaciones más estables [53].

El dendrograma de la figura 7.7, generado con los mejores hiperparámetros obtenidos para **HDBSCAN**, ofrece una visión de la agrupación de los datos según su densidad, desde clusters locales y pequeños hasta configuraciones más generales y menos densas. Esta representación jerárquica permite identificar patrones clave y resaltar la estructura interna de los grupos segmentados. La graduación de colores en el dendrograma, que va del rojo al azul, representa la variación en las distancias de fusión entre los clústers, mientras los tonos azules indican fusiones a distancias más cortas, los rojos corresponden a fusiones más lejanas.

En el eje y del dendrograma se representa el valor de λ o densidad inversa, que indica el nivel de densidad necesario para formar los clusters. Valores altos de λ corresponden a regiones más densas, mientras que valores bajos reflejan áreas menos densas. Este análisis jerárquico permitió identificar tres clases principales: dos clústeres relevantes, uno formado por la fusión de puntos a distancias cortas y otro generado a partir de puntos más dispersos, y un tercer grupo compuesto por puntos considerados como ruido, que no pudieron ser asignados a ninguno de los clústeres anteriores ni formar uno nuevo. El ruido es una característica inherente de **HDBSCAN**, ya que el algoritmo clasifica algunos datos como ruido cuando no cumplen con los criterios de densidad suficientes para formar un cluster estable [53].

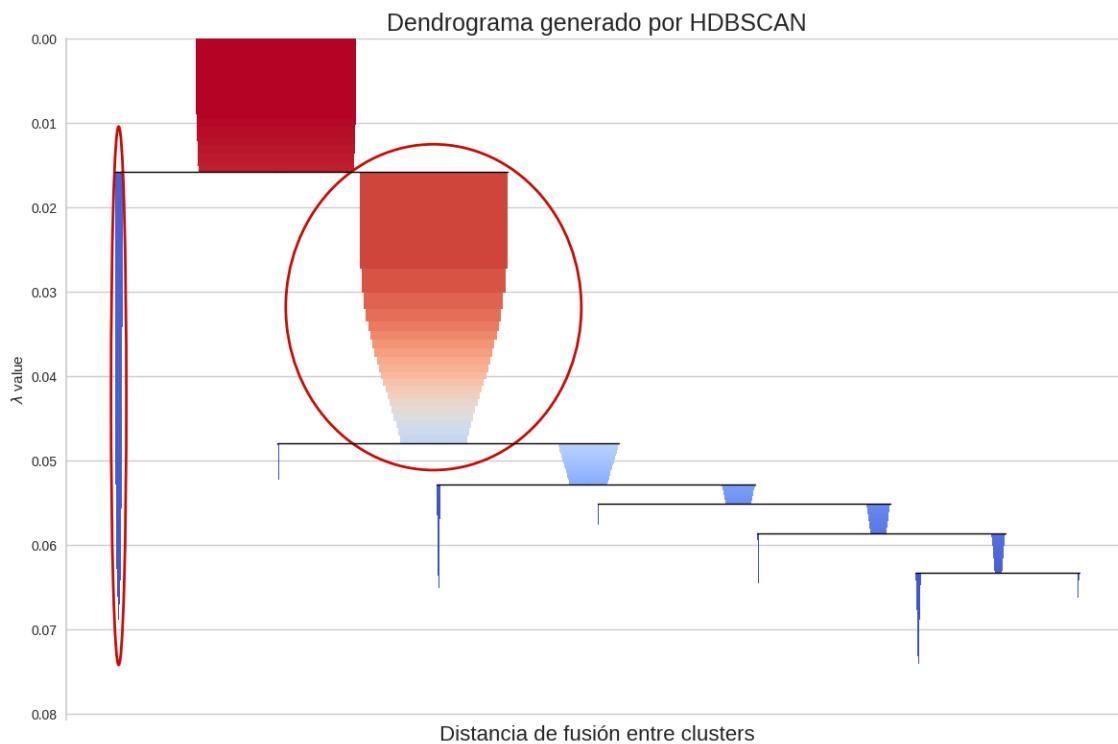


Figura 7.7: dendrograma y distribución de clústeres obtenidos mediante el algoritmo HDBSCAN.

A medida que disminuye el valor de λ , se distinguen dos ramas principales con longitudes significativas, lo que indica su robustez y persistencia a lo largo de diferentes niveles de densidad. Gracias a su capacidad para identificar clusters de diferentes formas y densidades, **HDBSCAN** ofrece una segmentación más flexible y precisa que otros algoritmos de clustering [53]. La primera rama principal se mantiene única, estable y compacta a lo largo de un amplio rango de λ , evidenciando un grupo robusto y significativo en los datos. La segunda rama por otro lado, muestra una reducción progresiva en el número de clusters a medida que disminuye λ , lo que sugiere una fragmentación en densidades más bajas. En un punto específico, esta rama se bifurca en dos subramas cortas que representan clusters más pequeños y menos relevantes, marcando el límite de su estabi-

lidad. Una de estas subramas, además, al disminuir el valor de densidad λ , continua subdividiéndose en varios clusters diferenciados aún más a lo largo de cuatro iteraciones.

Finalmente, el análisis del dendrograma confirma que los datos se estructuran principalmente en dos agrupaciones relevantes, claramente delimitadas visualmente mediante un círculo y una serie de puntos que corresponden a ruido. De estas dos agrupaciones principales, el clúster con etiqueta 1 sobresale como la agrupación más densa y dominante, al concentrar 23.885 observaciones. En contraste, el clúster con etiqueta 0 es considerablemente más pequeño, con 1.238 observaciones, lo que evidencia diferencias significativas en la distribución y densidad de los datos entre ambos grupos.

En cuanto a los datos sin clasificar en un cluster, el dendrograma también pone de manifiesto la capacidad del modelo para identificar regiones de baja densidad o datos atípicos, representados por los puntos asignados a la etiqueta -1, correspondientes a 717 observaciones que, al no presentar suficiente cohesión, no forman parte de ninguna agrupación seleccionada.

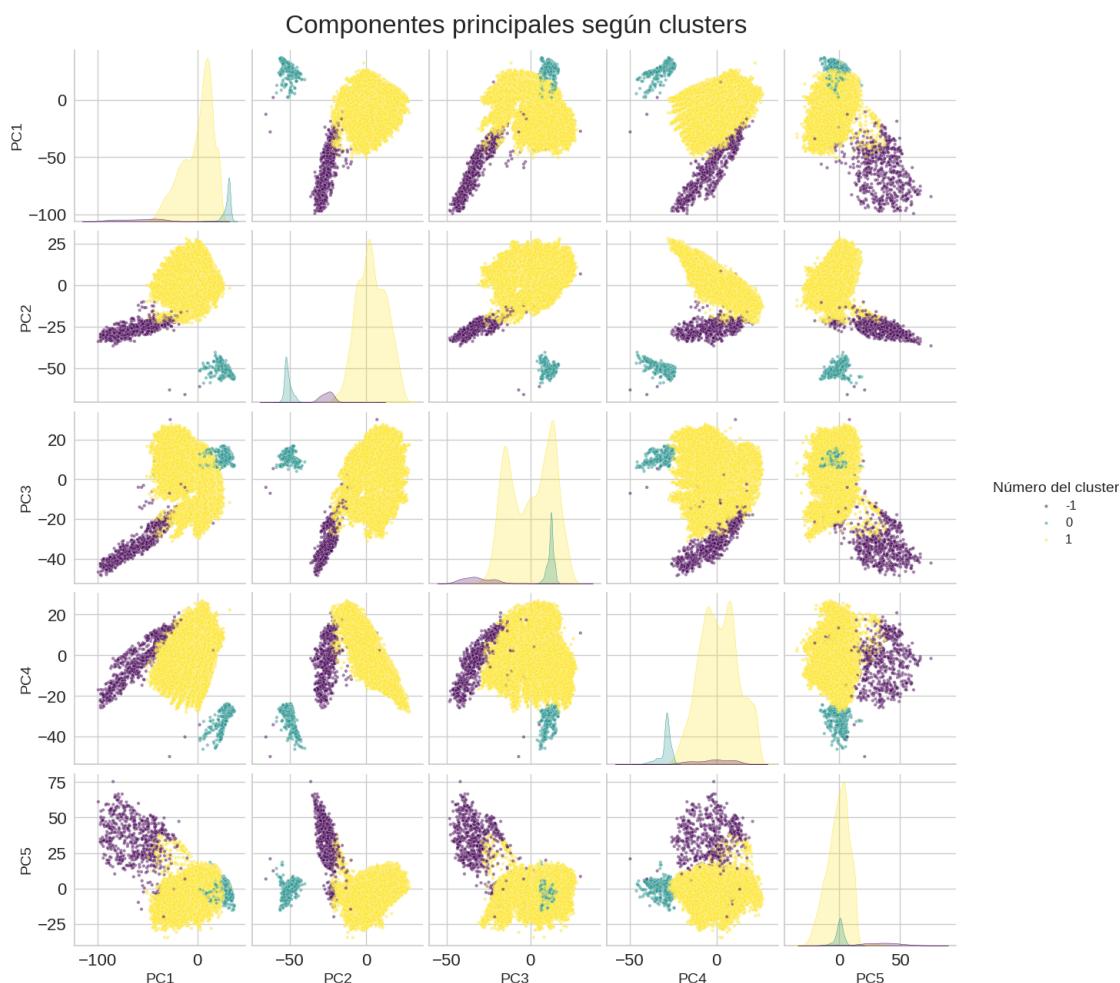


Figura 7.8: visualización de pares de las componentes extraídas, diferenciadas por los clústeres generados mediante HDBSCAN.

Al analizar la segmentación obtenida en el gráfico de las primeras componentes principales (figura 7.8), se observa una separación derivada de la interacción entre las componentes. Sin embargo, aunque existe una diferenciación clara, **HDBSCAN** no logra captar plenamente subgrupos a partir de las características disponibles, lo que pone de manifiesto ciertas limitaciones a la hora de generar una segmentación más precisa basada en la estructura subyacente de los datos.

7.3.2. K-Means

A diferencia de la aproximación utilizada para **HDBSCAN**, en el modelo **K-Means** no se aplicará una búsqueda bayesiana. En su lugar, como se muestra en el listing 7.16, se calcularán las métricas para un rango de valores predefinidos para el número de clústeres, de modo que el algoritmo explore directamente esas posibles particiones. Este enfoque busca identificar el número óptimo de clusters que ofrezcan los mejores resultados en términos de separación y homogeneidad de las clases.

```
Definir rango_de_clusters como el rango de 3 a 250
Función calcular_metricas_internas(datos, etiquetas):
    Calcular el índice de silueta (silhouette_avg) usando los datos y las etiquetas
    Calcular el índice Davies-Bouldin (davies_bouldin) usando los datos y las etiquetas
    Calcular el índice Calinski-Harabasz (calinski_harabasz) usando los datos y las
        etiquetas
    Retornar los tres índices: silhouette_avg, davies_bouldin, calinski_harabasz

Inicializar una lista vacía llamada resultados

Para cada k en rango_de_clusters:
    - Crear un objeto KMeans con k clusters, 10 inicializaciones y una semilla aleatoria
        de 42
    - Ajustar el modelo KMeans a los datos PCA_df
    - Calcular las métricas internas usando los resultados de KMeans:
    - silhouette_kmeans, davies_bouldin_kmeans, calinski_harabasz_kmeans =
        calcular_metricas_internas(PCA_df, etiquetas de KMeans)
    - Almacenar los resultados de las métricas en el formato de un diccionario:
        {"K": k,
        "Silhouette": silhouette_kmeans,
        "Davies-Bouldin": davies_bouldin_kmeans,
        "Calinski-Harabasz": calinski_harabasz_kmeans}

Convertir la lista de resultados en un DataFrame llamado results_df
```

Listing 7.16: cálculo de métricas de validación durante la búsqueda del número óptimo de clústeres para el modelo K-Means.

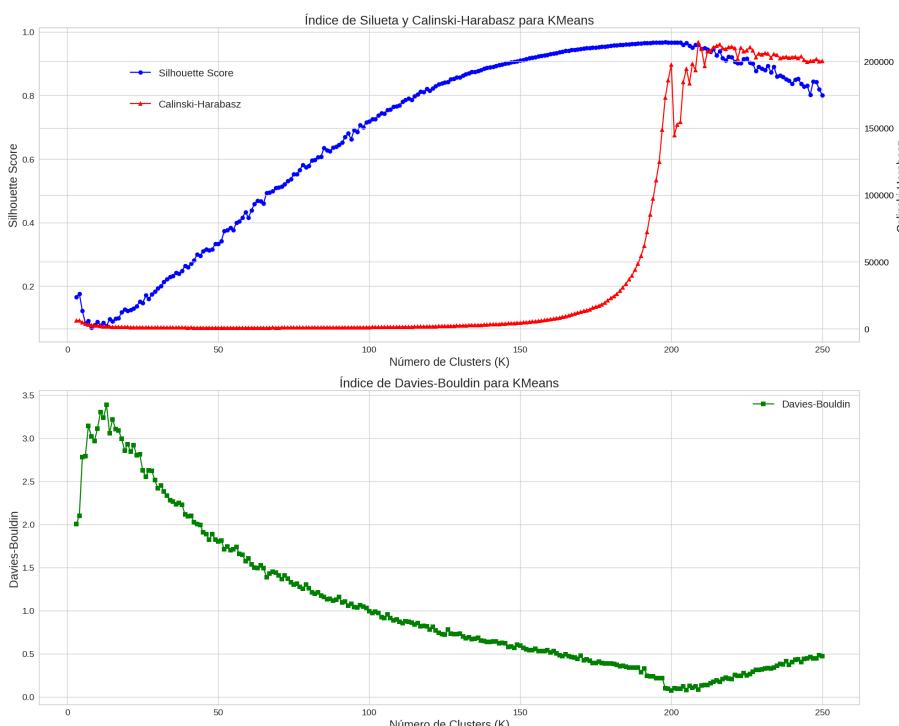


Figura 7.9: análisis de las métricas de validación en función del número de clústeres.

En la figura 7.9 se pueden visualizar las métricas resultantes para cada valor de k iterado. Dado que el rango de valores para el índice de Davies-Bouldin difiere considerablemente de las otras métricas, visualizamos sus resultados por separado.

Las métricas analizadas presentan comportamientos diferenciados a medida que aumenta el número de clústeres. El Silhouette score muestra un incremento constante, alcanzando su valor máximo de 0.96 alrededor de $k=200$.

Por su parte, el Calinski-Harabasz experimenta un descenso inicial seguido de una fase estable; sin embargo, a partir de $k=100$ aumenta rápidamente hasta $k=200$, donde comienza a oscilar con varios picos ascendentes y descendentes, alcanzando su punto máximo en $k=207$ con un índice de 214,286, tras lo cual disminuye de forma gradual.

En cuanto al Davies-Bouldin, se observa un aumento inicial hasta $k=20$, tras el cual el valor desciende de manera constante hasta alcanzar su mínimo de 0,08 en $k=203$. A partir de este punto, un incremento en el número de clusters provoca un aumento del índice, indicando una menor separación entre clases.

Considerando maximizar el desempeño de todas las métricas, buscando equilibrio entre densidad y separación de las etiquetas, se selecciona $k=200$ como número óptimo de clusters a considerar para **K-Means**.

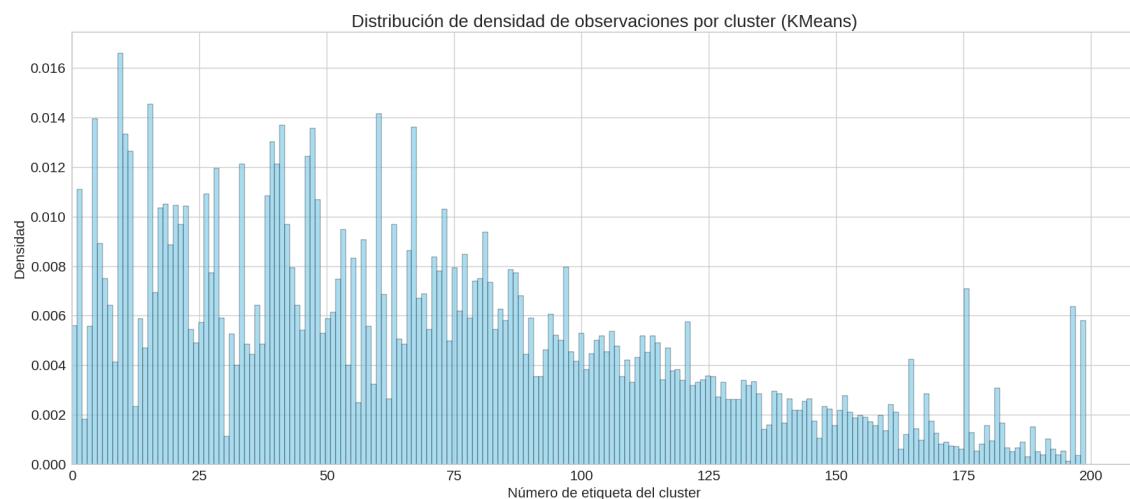


Figura 7.10: análisis de la distribución de observaciones por clúster.

El análisis de la frecuencia de asignación de etiquetas, mostrado en la figura 7.10, revela una notable variabilidad en los tamaños de los clusters. Algunas clases, como las 9, 15 y 50, agrupan más de 350 puntos, capturando patrones predominantes en los datos. En contraste, clusters como los 196, 188 y 198 tienen frecuencias mucho menores, con menos de 20 puntos cada uno, reflejando tendencias menos comunes. Sin embargo, no se observa ningún caso extremo que domine la distribución. Esta variabilidad evidencia la heterogeneidad inherente de los datos y puede ser valiosa para identificar subgrupos únicos en el conjunto de datos.

Al analizar en la figura 7.11, el gráfico de las componentes principales con las etiquetas asignadas por **K-Means**, se observa que ciertos clusters con etiquetas entre 160 y 200 tienden a concentrarse en valores bajos de las componentes principales PC1 y PC3. Asimismo, destaca un grupo diferenciado con valores altos en PC1 y bajos en PC2.

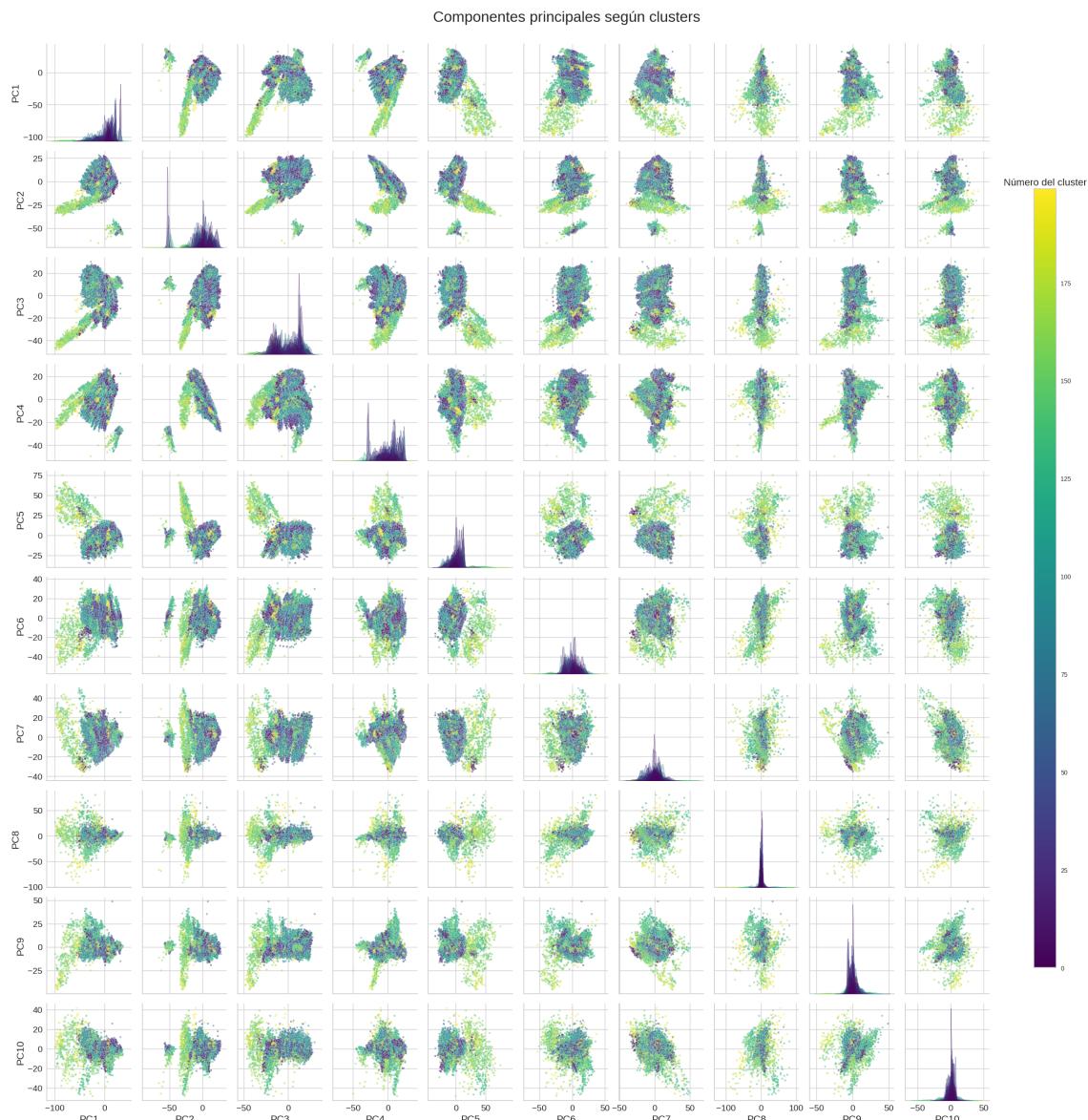


Figura 7.11: visualización de pares de las componentes principales, diferenciadas por los clústeres identificados mediante K-Means.

En general, las clases menos frecuentes, asignadas con etiquetas numéricas más altas, presentan una mayor dispersión y se sitúan más alejadas del centro de las distribuciones, donde sí se concentran las etiquetas con números más bajos. Estas observaciones, consideradas outliers, reflejan comportamientos inusuales en comparación con el resto. Esta detección de outliers está estrechamente vinculada al modelado de clusters, ya que mientras este método agrupa los datos según patrones mayoritarios, las anomalías representan casos excepcionales. En los enfoques basados en clustering, se consideran outliers aquellos datos que forman grupos pequeños y alejados o que no se ajustan a ningún cluster [32].

El elevado número de grupos dificulta la visualización de patrones claramente definidos y complica la interpretación de las estructuras más evidentes. Para abordar este desafío, se han analizado las etiquetas generadas para clusters de distinta densidad, distinguiendo entre las clases más representativas y las más atípicas. Este análisis se ha complementado con la aplicación de umbrales de frecuencia que filtran los clusters mostrados en las visualizaciones.

Al aplicar un filtro que considera únicamente las etiquetas de clusters con al menos 200 observaciones, se observa en la figura 7.12 una mejor separación en las primeras componentes principales. Sin embargo, en las últimas componentes, se evidencia un mayor solapamiento y dispersión, lo que sugiere una menor capacidad de estas variables para diferenciar eficazmente los grupos.

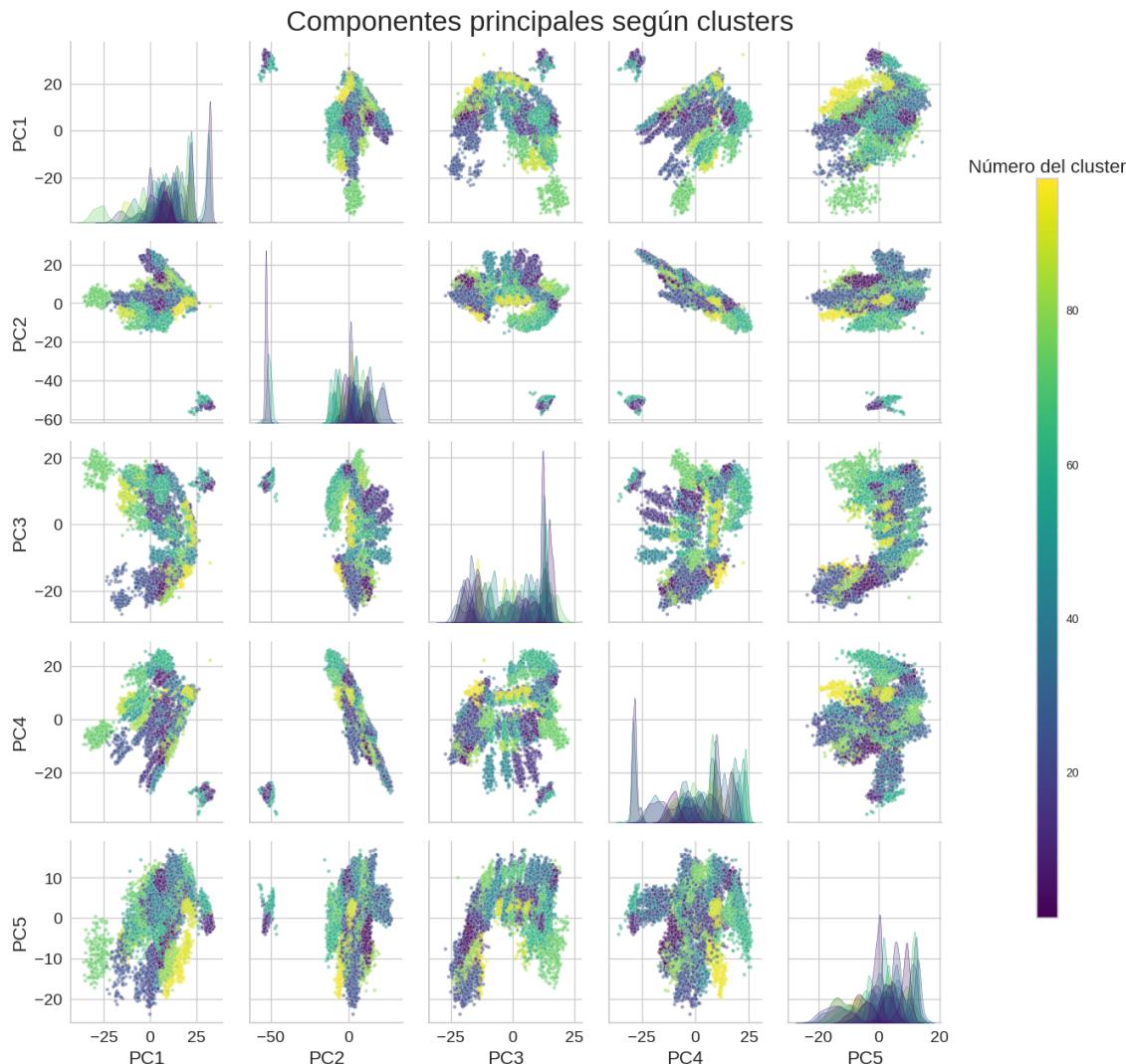


Figura 7.12: visualización de pares de las componentes principales, diferenciadas por los clústeres más frecuentes identificados mediante K-Means.

A continuación, se detallan las interacciones más relevantes:

- **Interacción PC1-PC2:** la distribución de puntos se concentra principalmente a lo largo del eje PC1, mientras que PC2 ofrece una separación más limitada. Dentro de esta interacción, se distinguen dos grupos según el valor de PC2: uno con valores bajos y otro con valores más elevados. En este segundo grupo, más numeroso, PC1 juega un papel clave en la diferenciación interna, mostrando una mayor uniformidad en valores bajos y una creciente diversidad de clusters a medida que aumenta su valor.
- **Interacción PC1-PC3:** esta combinación revela una mayor dispersión en ambas componentes, haciendo visibles algunos clusters que no se detectaban en la interacción PC1-PC2. Los puntos tienden a distribuirse a lo largo de PC1, destacando clusters específicos, como los de tonalidades verdes y violetas, que se separan notablemente del resto.

- **Otras componentes principales:** llama la atención la interacción entre las componentes 4 y 5, donde emergen grupos específicos, como el cluster amarillo, que no se distinguían claramente en las combinaciones anteriores.

Por otro lado, es fundamental analizar los clusters con menor frecuencia de aparición, ya que pueden representar patrones atípicos o anomalías relevantes para identificar comportamientos inusuales dentro del conjunto de datos. Para este propósito, se filtraron los grupos con menos de 200 observaciones.

Estos clusters muestran una mayor dispersión en general, como se puede observar en la figura 7.13, pero esta tendencia, como se mostró en la figura 7.11 se acentúa notablemente a medida que se avanza hacia la décima componente principal.

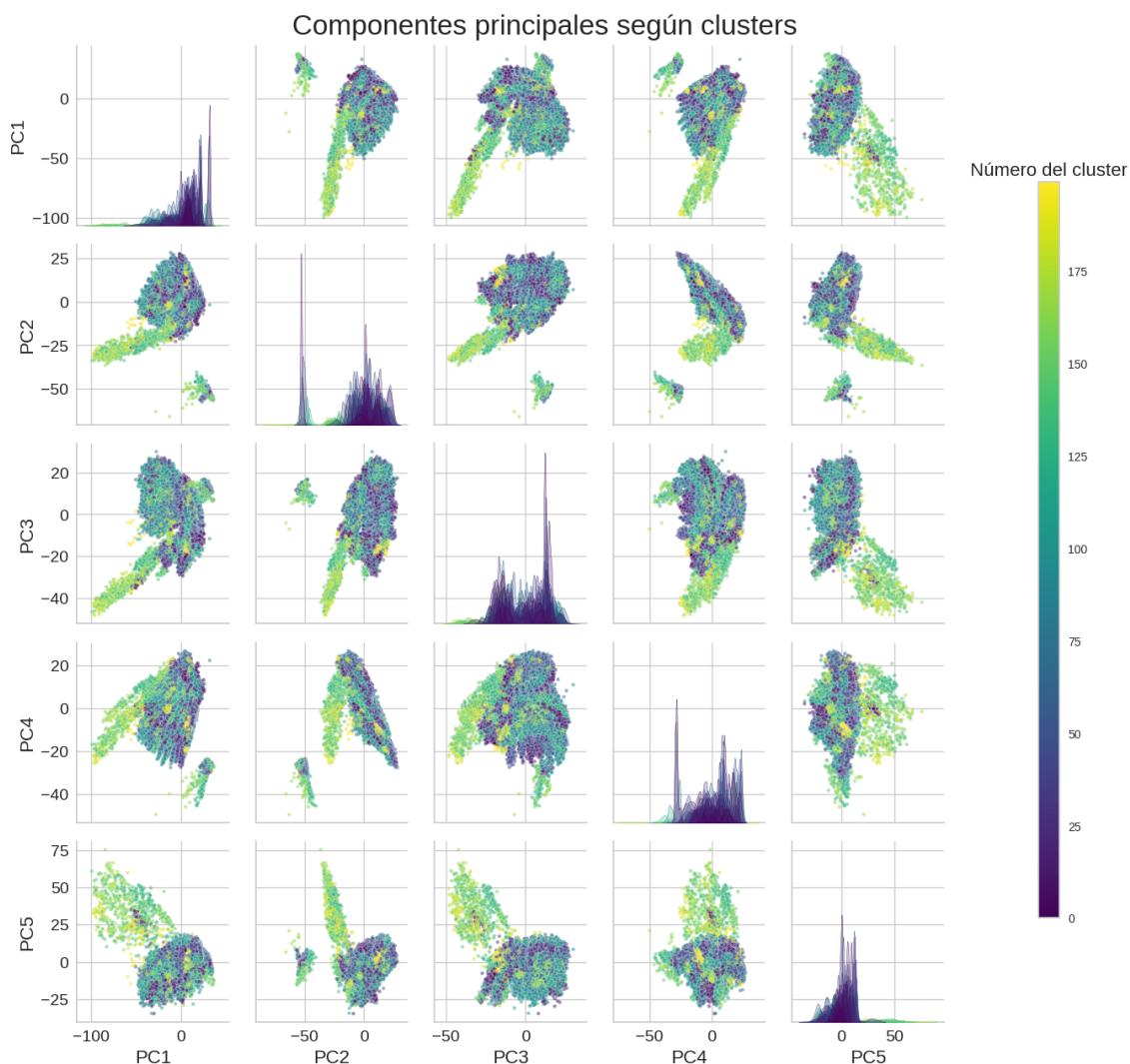


Figura 7.13: visualización del gráfico de las componentes principales, diferenciadas por los clústeres menos frecuentes identificados mediante K-Means.

Respecto a las clases, las primeras componentes, particularmente **PC2** y **PC4**, revelan una clara separación entre dos grupos de puntos: uno más numeroso, distribuido en valores elevados de estas componentes, y otro más pequeño, concentrado en valores bajos. Además, la distribución a lo largo del eje de **PC1** muestra una división adicional según si los valores son altos o bajos, destacando una segmentación más definida en comparación con los clusters más grandes. Aunque estos patrones coinciden con análisis previos de los clusters más frecuentes, los valores de las componentes son menos extremos, lo que

sugiere que no representan anomalías, sino clusters menores con características diferenciadas de los patrones transaccionales habituales [32].

7.4 Resultados y selección del modelo final de clustering

En la sección anterior se han evaluado dos enfoques de clustering, **HDBSCAN** y **K-Means**, aplicados sobre las componentes principales que capturan diferentes características de los intercambios analizados. El modelo **HDBSCAN**, si bien ofrece la capacidad de segmentar grupos de distinta densidad y permite detectar ruido, no alcanza el nivel de detalle requerido para el contexto de este estudio, limitándose a identificar únicamente tres agrupaciones, una de las cuales corresponde al ruido.

Por otro lado, el algoritmo **K-Means** ha mostrado una mayor capacidad de segmentación, generando un número más amplio de clusters. Algunos de estos grupos presentan una estructura bien definida y compacta, mientras que otros muestran una mayor dispersión. No obstante, las componentes principales han permitido diferenciarlos, confirmando que las características temporales, de frecuencia, tamaño, así como aquellas asociadas al tipo de mensaje, su destino y origen, y las configuraciones del módulo de seguridad, son suficientes para lograr una separación significativa entre los grupos utilizando este método.

De acuerdo con estos resultados y a las métricas de evaluación obtenidas, se concluye que **K-Means** proporciona una clasificación más adecuada y precisa para los datos analizados, permitiendo una segmentación que refleja de manera más fiel las diferencias en el comportamiento de las interacciones. Por tanto, se selecciona **K-Means** como modelo de clusterización para identificar patrones y comprender la dinámica y las características del flujo de mensajes de los clientes.

CAPÍTULO 8

Modelos de predicción

En el contexto del análisis de transmisiones EDI, anticipar el comportamiento futuro del sistema es fundamental para optimizar la planificación y garantizar la eficiencia operativa.

Dado que las transacciones se registran de manera continua a lo largo del tiempo, se ha adoptado un enfoque basado en series temporales, centrándose en el uso de Redes Neuronales Recurrentes (RNN), específicamente diseñadas para modelar dependencias temporales inherentes a este tipo de datos. Se espera que estos modelos permitan predecir con alta precisión la evolución del número de mensajes intercambiados en la plataforma, aprovechando patrones históricos para detectar tendencias, estacionalidades y posibles anomalías. Esta capacidad predictiva no solo enriquece la comprensión de las dinámicas de comportamiento, sino que también facilita la anticipación de picos de demanda, la gestión eficiente de recursos y la toma de decisiones estratégicas basadas en información fiable y actualizada.

Este capítulo aborda la preparación y adecuación del conjunto de datos para su uso en redes neuronales, así como la aplicación, entrenamiento y ajuste de modelos mediante la búsqueda de la combinación óptima de hiperparámetros y arquitecturas. El objetivo es maximizar la precisión de las predicciones a través del desarrollo de modelos capaces de capturar la naturaleza secuencial y dinámica del volumen de documentos transaccionados en el entorno del cliente.

8.1 Preparación de datos para el modelo predictivo

Para maximizar la efectividad del modelo predictivo, es fundamental seleccionar y preparar los datos de entrada adecuadamente, se han seleccionado exclusivamente las variables directamente relacionadas con la variable objetivo, así como aquellas que capturan dependencias temporales de tipo cíclico o con efectos de rezago. Este enfoque busca reducir el número de características sin perder la información esencial para modelar patrones complejos necesarios en la predicción, enfatizando la importancia de seleccionar características relevantes para reducir la sobrecarga de información y optimizar el rendimiento del modelo [64]. Es fundamental que las variables agregadas complementen la información existente en lugar de sustituirla, permitiendo preservar características clave y mitigando el problema del gradiente que se desvanece [43].

Para la preparación de los datos, el primer paso consistió en agrupar los mensajes por patrones horarios, tal como se muestra en el Listado 7.1, obteniendo así el número de mensajes intercambiados en cada periodo.

1. Agrupar los datos (df) por la columna "`dateInsert`" con una frecuencia de una hora.
 2. Para cada grupo de datos basado en la fecha y hora ("`dateInsert`"), Asignar a nueva columna "`msg_num`" conteo el tamaño de cada grupo (número de filas por hora)

Listing 8.1: agregación de los datos a una resolución temporal horaria.

Las series temporales se caracterizan por movimientos cílicos, variaciones estacionales y tendencias, que reflejan su evolución a lo largo del tiempo [32] por lo que es importante comprender las tendencias y patrones estacionales que puedan existir para la variable

objetivo. En la figura 8.1 se observa la evolución del número de mensajes transaccionados por periodo, representada en la gráfica de tendencia correspondiente a la descomposición temporal de la serie. Si bien existen picos de crecimiento y de decrecimiento, la tendencia general es al alza y constante, dejando claro que el volumen del flujo EDI del cliente es creciente.

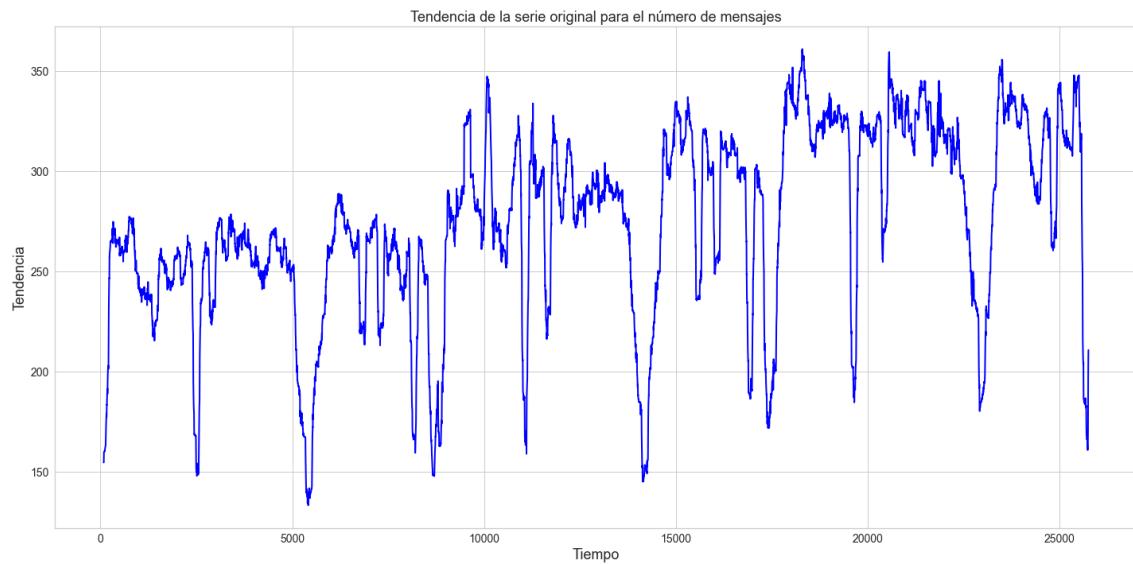


Figura 8.1: análisis de la tendencia del volúmen de transacciones.

Por otro lado, la figura 8.2 muestra la descomposición estacional del número de mensajes. En este análisis, los valores negativos indican períodos de menor actividad respecto al promedio, mientras que los valores positivos reflejan una actividad superior a la esperada. El patrón estacional exhibe una estructura bien definida, con picos marcados y recurrentes, lo que sugiere una periodicidad semanal. Se observa además, que el volumen de transmisiones es relativamente bajo durante los tres primeros días de la semana, aumentando significativamente en los cuatro días siguientes.

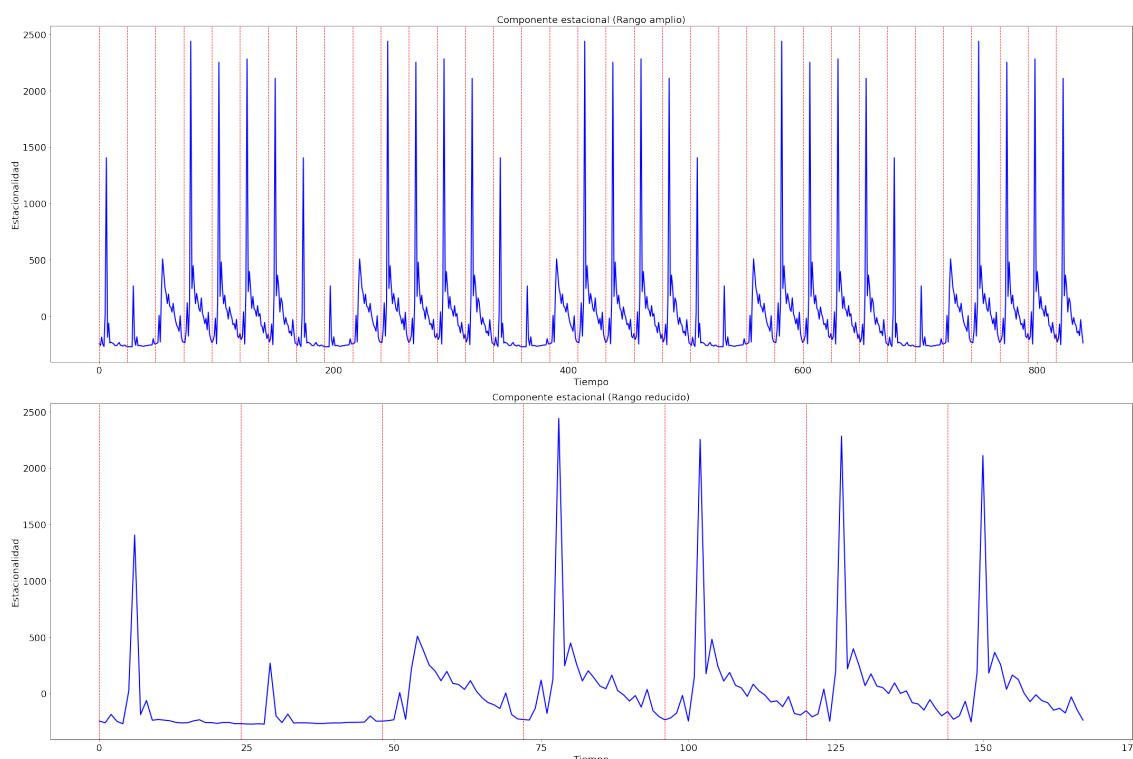


Figura 8.2: análisis de la estacionalidad del volumen de transacciones.

Al analizar la estacionalidad semanal, tomando el sábado como inicio del ciclo, se aprecia un patrón atípico con una actividad elevada durante los fines de semana, especialmente los sábados. En contraste, los domingos y lunes presentan los niveles más bajos de transacciones. El martes alcanza el pico máximo de actividad, seguido de una recuperación notable que, a partir de ese día, disminuye progresivamente hacia el final de la semana.

A nivel diario, se revela que la actividad es más intensa al comienzo del día y decrece de forma gradual, aunque se registran algunos picos aislados a lo largo de este descenso.

La incorporación de características temporales como festivos, día, hora, mes y estacionalidad en los modelos predictivos mejora la precisión y facilita la identificación de patrones ocultos. Esto no solo permite una mejor comprensión de los datos, sino que también optimiza el rendimiento de los modelos en contextos donde el factor temporal influye en los resultados [47]. Tras identificar patrones estacionales a nivel horario, mensual y semanal, se introdujeron según muestra el listing 8.2, variables cíclicas temporales para capturar de manera más efectiva las dependencias temporales en el análisis. Además, considerando su impacto en la diferenciación de patrones transaccionales, se añadió una variable binaria para identificar si una observación ocurre durante el fin de semana.

```

1. Crear variables cíclicas para la hora del día:
Para cada fila en "dateInsert":
    hour_sin = sin(2 * π* (hora de "dateInsert") / 24)
    hour_cos = cos(2 * π* (hora de "dateInsert") / 24)

2. Crear variables cíclicas para el día de la semana:
Para cada fila en "dateInsert":
    day_of_week_sin = sin(2 * π* (día de la semana de "dateInsert") / 7)
    day_of_week_cos = cos(2 * π* (día de la semana de "dateInsert") / 7)

3. Crear variables cíclicas para el mes:
Para cada fila en "dateInsert":
    month_sin = sin(2 * π* (mes de "dateInsert") / 12)
    month_cos = cos(2 * π* (mes de "dateInsert") / 12)

4. Crear variables cíclicas para la semana del año:
Para cada fila en "dateInsert":
    week_of_year_sin = sin(2 * π* (semana ISO del año de "dateInsert") / 12)
    week_of_year_cos = cos(2 * π* (semana ISO del año de "dateInsert") / 12)

5. Identificar si el día es fin de semana:
Para cada fila en "day_of_week":
    Si el día de la semana es 5 (sábado) o 6 (domingo):
        is_weekend = 1
    En caso contrario:
        is_weekend = 0

```

Listing 8.2: cálculo de variables cíclicas temporales.

De igual forma, considerando la relación entre rezagos observada en la figura 7.1, como se muestra en el listing 8.3 se han incorporado, aquellos que resultaron más significativos (**168, 336, 504 y 672**) derivados del número de mensajes intercambiados. Esto permite que los modelos capturen mejor las relaciones temporales, ya que las variables rezagadas utilizan valores previos de la serie para reflejar patrones de estacionalidad y tendencias, mejorando así la precisión de las predicciones [47].

```

Función generar_características_con_lags(df, columnas, lags):
    Para cada columna en columnas:
        Para cada lag en lags:
            Convertir el lag a entero
            Crear una nueva columna en df_features con el nombre "columna_lag_lag" y
                asignar los valores desplazados de la columna correspondiente
            Rellenar los valores nulos con 0 en la nueva columna
        Retornar conjunto
    # Llamar a la función con la columna "msg_num" y los lags [168, 336, 504, 672]

```

Listing 8.3: agregación de los rezagos temporales más significativos en el número de mensajes transaccionados.

Tras el preprocessamiento, contamos con 25.840 observaciones y 14 variables, incluyendo variables temporales cíclicas y el número de mensajes, junto con los rezagos calculados.

8.1.1. División de datos para el modelo predictivo

Dividir el conjunto de datos en entrenamiento, test y validación es fundamental para desarrollar modelos de clasificación y predicción robustos y generalizables. El conjunto de entrenamiento permite al modelo aprender patrones y relaciones, mientras que el conjunto de validación ayuda a ajustar los hiperparámetros y evaluar el desempeño durante la optimización. Finalmente, el conjunto de prueba se utiliza para evaluar el rendimiento del modelo frente a datos no vistos previamente. Esta separación previene el sobreajuste, garantiza una evaluación imparcial y optimiza el modelo al facilitar su ajuste [48]. Dado el volumen adecuado de transacciones disponible, en el listing 8.4 se va a destinar el 80 % del conjunto al entrenamiento, el 10 % a la validación y el 10 % restante a la prueba. Esta partición garantiza un equilibrio entre flexibilidad y relevancia en el análisis y desarrollo del modelo. Como resultado, se obtienen tres conjuntos de datos: el conjunto de entrenamiento, con 20.337 observaciones y 14 columnas, y dos conjuntos adicionales, validación y prueba, cada uno con 2.249 observaciones y 14 columnas.

```

1. Definir proporciones para los conjuntos:
    tamaño_entrenamiento = 0.8
    tamaño_validación = 0.1
    tamaño_prueba = 0.1
2. Calcular el número de filas para cada conjunto:
    filas_entrenamiento = redondear(tamaño_entrenamiento * total_filas_dataset)
    filas_validación = redondear(tamaño_validación * total_filas_dataset)
3. Dividir el dataset en los tres conjuntos:
    conjunto_entrenamiento = dataset[desde el inicio hasta filas_entrenamiento]
    conjunto_validación = dataset[desde filas_entrenamiento hasta filas_entrenamiento +
        filas_validación]
    conjunto_prueba = dataset[desde filas_entrenamiento + filas_validación hasta el final]
```

Listing 8.4: división de los datos en conjunto de entrenamiento, validación y prueba.

Tras la división del conjunto de datos, para facilitar la convergencia óptima y mejorar el aprendizaje del modelo, es fundamental garantizar que todas las características tengan un impacto equilibrado, independientemente de su rango. Cuando los predictores numéricos presentan escalas o varianzas diferentes, pueden influir desproporcionadamente en el modelo, afectando su interpretación y rendimiento. Para mitigar este problema, se aplica un proceso de normalización que transforma los datos a una escala común, centrándolos y dividiéndolos por su desviación típica. Esto no solo equilibra la influencia de cada predictor, sino que también mejora la estabilidad y precisión del modelo [49].

El escalado debe realizarse utilizando exclusivamente las métricas calculadas a partir del conjunto de entrenamiento y aplicarlas posteriormente a los datos de validación y prueba, tal y como se realiza en el listing 8.5. Estandarizar todo el conjunto antes de la división podría provocar una fuga de información, ya que el modelo accedería indirectamente a datos de prueba durante el entrenamiento. Esto podría generar una percepción engañosa de un buen rendimiento en la evaluación, pero afectaría negativamente su capacidad de generalización a nuevos datos [50].

```

1. Inicializar el escalador y calcular las métricas de escalado usando los datos de
    entrenamiento:
    scaler = StandardScaler()
```

```

scaled_numerical = scaler.fit_transform(train_df)
2. Escalar los datos de entrenamiento, validación y prueba utilizando las métricas del
   escalador.
Train_scaled = scaler.transform(train_df)
Val_scaled = scaler.transform(validation_df)
Test_scaled = scaler.transform(test_df)

```

Listing 8.5: escalado de los conjuntos de datos.

Para alimentar una red neuronal con datos secuenciales, es esencial ajustar el formato de entrada al modelo. Una vez escalados los conjuntos de datos, se generan tensores mediante ventanas temporales deslizantes, lo que permite realizar predicciones con redes neuronales. Un tensor es una estructura de datos que generaliza el concepto de matriz a múltiples dimensiones. En el caso de datos secuenciales, se emplea un tensor tridimensional, donde el segundo eje (índice 1) representa la dimensión temporal. Esta organización optimiza tanto el almacenamiento como el procesamiento eficiente de secuencias en modelos de aprendizaje profundo [51]. Considerando la alta correlación y los rezagos analizados en la figura 7.1, se establecen ventanas temporales y un horizonte de predicción de 168 períodos, como se observa en el listing 8.6.

```

CONSTANTE PREDICTION_HORIZON = 168
CONSTANTE WINDOW_SIZE = 168

Función df_a_X_y_multioutput(df, columnas_objetivo, tamaño_ventana=WINDOW_SIZE,
    horizonte_predicción=PREDICTION_HORIZON):
    Convertir df en un array de numpy y asignarlo a df_as_np

    Obtener los índices de las columnas objetivo:
        Para cada columna en columnas_objetivo:
            Obtener la posición de la columna en df y agregarla a target_indices

    Extraer las variables objetivo (y) desde df_as_np usando los índices de columnas
        objetivo

    Inicializar listas vacías X_data y y_data

    Generar secuencias de ventanas temporales:
        Para cada índice i en el rango de (longitud de df_as_np - tamaño_ventana -
            horizonte_predicción + 1):
            Agregar la ventana de tamaño_ventana de df_as_np a X_data
            Agregar las variables objetivo específicas (horizonte_predicción) a y_data

    Retornar np.array(X_data) y np.array(y_data)

# Llamar a la función con las columnas objetivo "msg_num"
columnas_objetivo = ["msg_num"]

```

Listing 8.6: conversión de los datos en tensores con ventanas deslizantes para su uso como entrada del modelo.

Mediante este proceso, se generan tensores con secuencias de 168 registros consecutivos, donde las características de entrada (**X**) tienen dimensiones **(20337, 168, 14)** para el conjunto de entrenamiento y **(2249, 168, 14)** para los conjuntos de prueba y validación. Las etiquetas de salida (**y**), correspondientes a la variable objetivo, presentan dimensiones **(20337, 168, 1)** para el entrenamiento y **(2249, 168, 1)** para prueba y validación. Estos tensores permiten capturar la estructura temporal de los datos y entrenar el modelo de manera efectiva.

Con el objetivo de depurar el diseño, evaluar el entrenamiento y optimizar un modelo de red neuronal para predecir el número de mensajes transaccionados. No solo es importan-

te lograr precisión de las predicciones sobre los datos de validación, sino también garantizar una adecuada generalización a observaciones no vistas durante el entrenamiento. Un sobreajuste podría introducir sesgo y reducir la eficacia del modelo en nuevos escenarios [46]. Para evaluar la calidad de las predicciones se tendrán en cuenta las siguientes métricas de validación:

- *Mean squared error (MSE)*: la diferencia al cuadrado entre los valores predichos y reales previene la cancelación de términos negativos y asigna mayor peso a los errores grandes, asegurando un descenso de gradiente hacia un único mínimo global. Sin embargo, su alta sensibilidad a los valores atípicos puede amplificar significativamente los errores. Un valor cercano a 0 refleja una alta precisión, indicando que las predicciones se ajustan estrechamente a los valores reales. Por otro lado, la raíz cuadrada del MSE *Root Mean Squared Error (RMSE)*, representa la magnitud promedio de los errores y puede ser útil para evaluar el entrenamiento del modelo. Es más intuitivo y fácil de interpretar, ya que se expresa en las mismas unidades que la variable de salida. Sin embargo, mantiene las mismas limitaciones que el MSE, siendo sensible a los errores grandes [58].
- *Mean absolute error (MAE)*: mide la diferencia promedio entre las predicciones y los valores reales sin elevar los errores al cuadrado, lo que reduce su sensibilidad a valores atípicos. Su interpretación es sencilla, ya que proporciona una medida coherente del rendimiento del modelo en la misma escala que la variable objetivo. Un valor cercano a cero refleja una alta precisión en las predicciones [58].
- *Coeficiente de determinación (R²)*: el R², a diferencia de otras métricas que miden errores absolutos o relativos, refleja el desempeño global del modelo al indicar la proporción de variabilidad explicada en la variable dependiente. Un valor cercano a 1 sugiere un buen ajuste, aunque no necesariamente garantiza la corrección del modelo ni la ausencia de sesgos en los datos [58].

Durante el entrenamiento de redes neuronales, se busca minimizar la función de pérdida ajustando los parámetros del modelo para mejorar su desempeño [46]. En este caso, el error cuadrático medio (MSE) se emplea como métrica principal en la búsqueda bayesiana para comparar modelos y seleccionar la configuración óptima, debido a su sensibilidad ante errores de gran magnitud.

8.2 Diseño del modelo

Una red neuronal recurrente es una red con conexiones dirigidas a lo largo del tiempo, lo que le permite procesar secuencias y retener información en su estado interno [45].

Para capturar patrones temporales complejos y modelar relaciones no lineales en los datos, es esencial diseñar una arquitectura óptima que combine adecuadamente sus capas. La figura 7.3 ilustra de forma sencilla la estructura general de una red neuronal con capas intermedias. En este contexto, las celdas de modelos como **LSTM** y **GRU** incorporan además mecanismos de entrada, olvido y salida, que regulan dinámicamente la memoria mediante una estructura modular. La optimización del desempeño en la predicción de series temporales depende, además, del ajuste de hiperparámetros clave, como las funciones de activación y los pesos de las conexiones recurrentes [46].

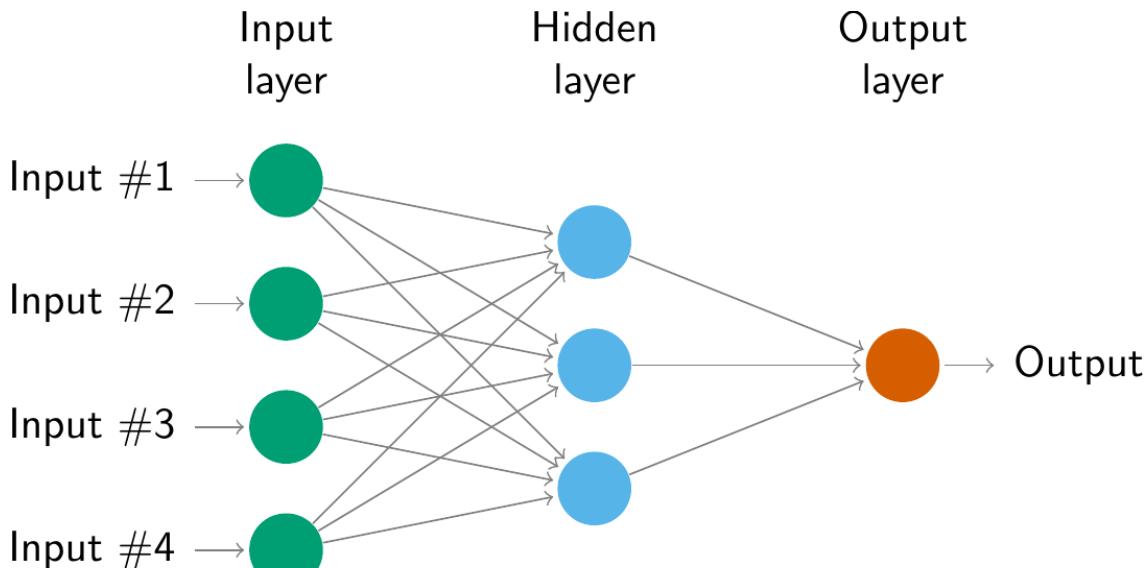


Figura 8.3: gráfica extraída de Hyndman and Athanasopoulos, 2018 [67]. Que muestra una red neuronal con cuatro entradas y una capa oculta con tres neuronas.

Capa de entrada (InputLayer):

La correcta definición de la forma de entrada del modelo es fundamental. Basado en el análisis de la autocorrelación de rezagos en 168 períodos y la detección de una estacionalidad semanal (figura 7.1), se establecieron ventanas temporales de 168 períodos para las 14 variables de entrada, dando lugar a una estructura tensorial de forma **(168, 14)**, utilizada en la red neuronal recurrente.

Capa GRU/ LSTM:

Orientadas a capturar patrones en series temporales. Son variantes de RNN que ayudan a modelar dependencias temporales en los datos.

- *units_encoder*: número de unidades de la capa, que determinan la capacidad de aprender patrones complejos.
- *activation_encoder*: la función de activación es fundamental en estas capas, ya que determina cómo se procesan las señales internas y se transforman las entradas en salidas, introduciendo la no linealidad necesaria en el modelo. Entre las opciones a considerar se incluyen:
 - *tanh (Tangente hiperbólica)*: escala los valores entre -1 y 1, centrando la salida en cero, lo que ayuda a normalizar las señales internas y prevenir la explosión o desaparición de gradientes, equilibrando así estabilidad y rendimiento.
 - *swish*: ofrece una salida suave y no lineal, facilitando al modelo el aprendizaje en problemas complejos o con alta no linealidad.
 - *relu (Rectified Linear Unit)*: anula los valores negativos y permite el paso de los positivos, siendo computacionalmente eficiente y evitando problemas de saturación.
 - *elu (Exponential Linear Unit)*: es una versión mejorada de ReLU que incorpora una curva suave en los valores negativos, lo que ayuda a mitigar la desaparición de gradientes y resulta especialmente útil cuando los datos contienen numerosas señales negativas.
- *l2_regularization_encoder*: regularización L2 aplicada a los pesos de la capa ayudando a prevenir el sobreajuste.

- *Dropout*: tasa de abandono para prevenir el sobreajuste apagando aleatoriamente algunas neuronas durante el entrenamiento.

Capa de atención:

Un avance fundamental en redes neuronales es la incorporación de mecanismos de atención, que permiten al modelo enfocar dinámicamente distintas partes de la secuencia de entrada en cada paso del proceso de generación de salida. Este enfoque ha demostrado ser especialmente útil en tareas como la traducción automática y en modelos con memoria explícita, al facilitar la propagación de información relevante a lo largo del tiempo [46].

El mecanismo de Atención Multi-Cabeza implementado en el modelo **Transformer**, introducido por Vaswani et al.[56], destaca por modelar dependencias globales en una sola pasada utilizando únicamente capas de atención, a diferencia de **LSTM** y **GRU**, que procesan datos de forma secuencial. Sin embargo, la combinación de **LSTM/ GRU** con atención ofrece una solución híbrida que integra la memoria recurrente de los modelos con la capacidad de destacar las partes más relevantes de la secuencia. Este enfoque, inspirado en los **Transformers**, permite que múltiples cabezas de atención capturen simultáneamente patrones temporales distintos, mejorando la capacidad de modelar relaciones complejas en secuencias temporales donde algunas partes de los datos son más significativas que otras. Además, esta combinación sigue siendo eficiente para el procesamiento secuencial debido a su menor costo computacional [46].

La propuesta de integrar atención por bloques en RNN ha demostrado mayor flexibilidad y capacidad de generalización, con resultados positivos en diversos estudios [54], [55].

- *num_heads_attention*: número de cabezas de atención que procesan los datos de forma independiente para luego combinarse, permitiendo al modelo capturar diversos tipos de relaciones en los datos.
- *key_dim_attention*: dimensión de las claves y valores de atención, que determina la cantidad de información que cada cabeza de atención es capaz de procesar.
- *dropout_attention*: tasa de abandono específica sobre la atención, ayudando a prevenir sobreajuste en esta capa.

Capa de normalización de lote. (BatchNormalization):

Propuesta por Ioffe y Szegedy en 2015, es una técnica que estabiliza y acelera el entrenamiento de redes neuronales al normalizar dinámicamente la media y la varianza de cada lote de datos, utilizando un promedio móvil para capturar cambios durante el proceso de entrenamiento. Su principal ventaja es mejorar la propagación del gradiente, mitigando problemas como su desaparición o explosión, lo que facilita la optimización y permite entrenar redes más profundas de manera eficiente. Esta técnica resulta especialmente útil en modelos complejos, ya que promueve una mayor estabilidad durante el entrenamiento, ayudando a prevenir fluctuaciones inesperadas en el proceso de aprendizaje [51].

Capa Densa (Dense):

La representación final antes de la salida pasa por una capa densa con activación ReLU, la cual introduce no linealidades al modelo y permite aprender representaciones más complejas antes de generar la predicción para los 168 períodos de la serie temporal [66].

- *Dense_units*: número de unidades que definen la capacidad de la capa para modelar relaciones no lineales.

- *Dropout_rate*: tasa de Dropout para capa densa ayudando a prevenir el sobreajuste.

Capa de salida (Dense):

Formada por una sola neurona con activación lineal, que se utiliza para predecir valores continuos.

No existe una regla universal para determinar el número ideal de capas ocultas en una red neuronal recurrente, ya que factores como la complejidad de los datos y el riesgo de sobreajuste influyen significativamente en el rendimiento durante el entrenamiento y las pruebas. Por ello, la experimentación con distintas arquitecturas y configuraciones resulta fundamental para encontrar la opción más adecuada [46]. Cada capa puede ser crucial para garantizar un flujo eficiente de información y mejorar la capacidad del modelo para centrarse en los aspectos más relevantes y minimizar problemas de generalización. El listing 8.7 presenta la configuración global de las posibles capas consideradas durante la búsqueda bayesiana, proceso que se explica en la sección 5.2.

```

FUNCIÓN construir_modelo(hp, modelo_preentrenado=None)
    INICIALIZAR modelo COMO Sequential()

    AÑADIR CAPA DE ENTRADA InputLayer A modelo CON input_shape=(168, 14)

    AÑADIR CAPA GRU A modelo CON:
        unidades = hp.Int("units_lstm_encoder", RANGO 100 A 700, PASO 50)
        activación = hp.Choice("activation_encoder", OPCIONES ["tanh", "swish", "elu",
            "relu"])
        return_sequences = Verdadero
        regularizador_kernel = 12(hp.Float("l2_regularization_encoder", RANGO 1e-5 A 1e-3,
            PASO 1e-5))
        dropout = hp.Float("dropout_lstm", RANGO 0.2 A 0.4, PASO 0.1)

        # Parámetros de la capa de atención
        DEFINIR num_cabezas = hp.Int("num_heads_attention", RANGO 2 A 8, PASO 2)
        DEFINIR dim_clave = hp.Int("key_dim_attention", RANGO 32 A 128, PASO 32)
        DEFINIR tasa_dropout_atención = hp.Float("dropout_attention", RANGO 0.1 A 0.5, PASO
            0.1)

        # Bloque de atención personalizado
        AÑADIR CustomAttentionBlock A modelo CON:
            num_cabezas = num_cabezas
            dim_clave = dim_clave
            tasa_dropout = tasa_dropout_atención

        # Normalización por lotes
        AÑADIR BatchNormalization A modelo

        # Capa densa
        AÑADIR Dense A modelo CON:
            unidades = hp.Int("dense_units", RANGO 128 A 512, PASO 32)

        # Capa de activación
        AÑADIR Activación A modelo CON activación = "relu"

        # Capa de Dropout
        AÑADIR Dropout A modelo CON:
            tasa = hp.Float("dropout_rate", RANGO 0.1 A 0.5, PASO 0.1)

        # Capa de salida
        AÑADIR Dense A modelo CON:
            unidades = 1
            activación = "linear"

```

```
RETORNAR modelo
```

Listing 8.7: definición de la arquitectura de la red neuronal y del espacio de búsqueda de hiperparámetros para la optimización bayesiana.

8.2.1. Búsqueda de hiperparámetros

La optimización de hiperparámetros se llevará a cabo mediante búsqueda bayesiana (como se explica en la sección 5.2). Para su ejecución, es fundamental definir ciertos hiperparámetros clave, como se muestra en el listing 8.8, que contribuyan a reducir el coste computacional:

- *max_trials*: número máximos de pruebas que se realizan para encontrar la mejor combinación.
- *executions_per_trial*: número de veces que se entrena el modelo con cada combinación para obtener una estimación más robusta de su rendimiento.
- *objective*: el objetivo de la optimización es minimizar la pérdida de validación, definida mediante la métrica de error cuadrático medio (MSE).
- *epochs*: cada época representa una pasada completa por el conjunto de datos de entrenamiento. Un mayor número de ellas permite al modelo aprender patrones más complejos, pero un exceso puede hacer que el aprendizaje se estabilice, de modo que continuar el entrenamiento más allá de ese punto no genera mejoras significativas.
- *batch_size*: número de muestras procesadas simultáneamente antes de actualizar los pesos del modelo. Es importante encontrar un equilibrio entre lotes pequeños, que pueden capturar mejor la variabilidad de los datos pero son más costosos computacionalmente, y lotes más grandes, que ofrecen mayor eficiencia pero pueden comprometer la capacidad de generalización.

```
INICIALIZAR afinador COMO BayesianOptimization CON:
    modelo_construcción = construir_modelo
    objetivo = "val_loss"
    pruebas_máximas = 30 # Número de combinaciones a probar
    ejecuciones_por_prueba = 3 # Número de ejecuciones por combinación de hiperparámetros

LLAMAR tuner.search CON:
    datos_entrenamiento = (X_train, y_train)
    epochs = 100 # Número de épocas de entrenamiento
    batch_size = 128
    datos_validación = (X_val, y_val) # Conjunto de datos para validación
    callbacks = [early_stopping, lr_scheduler] # Lista de callbacks para optimización
```

Listing 8.8: definición de parámetros y ejecución de la búsqueda bayesiana.

Durante el proceso de optimización, cuyo objetivo es minimizar el MSE, es fundamental seleccionar tanto un optimizador como una tasa de aprendizaje apropiados, tal como se ilustra en el listing 8.9, donde se evalúan distintas combinaciones de estos parámetros para determinar su impacto en el rendimiento del modelo. El optimizador define el método para actualizar los parámetros mediante el gradiente de la pérdida [51], mientras que la tasa de aprendizaje controla la magnitud de esos ajustes [59], afectando directamente la estabilidad y velocidad de convergencia del modelo. Una tasa alta acelera las actualizaciones, pero puede provocar inestabilidad, mientras que una tasa baja promueve una convergencia más estable a costa de un mayor tiempo de entrenamiento. Asimismo, el

decaimiento gradual de la tasa de aprendizaje permite un refinamiento progresivo del modelo, mejorando su convergencia final [59]. A continuación, se presentan las variables definidas para el proceso de optimización del modelo, con el objetivo de identificar la mejor combinación de hiperparámetros durante la búsqueda bayesiana:

- *Loss*: mediante MSE para evaluar el desempeño.
- *Optimizer*: diferentes algoritmos como Adam, AdamW, RMSprop, Adamax y Nadam, los cuales difieren en la forma de actualizar los pesos y gestionar el aprendizaje, afectando tanto la velocidad de convergencia como el rendimiento final del modelo.
- *Learning_rate*: tasa de aprendizaje que controla lo rápido que el optimizador ajusta los pesos.

```
#Dentro de la función construir_modelo
DEFINIR optimizer elección entre ["adam", "adamw", "rmsprop", "adamax", "nadam"]

DEFINIR learning_rate como un valor en el rango [1e-5, 5e-4] con pasos de 5e-5

SI optimizer = "adamw":
    ASIGNAR optimizer como AdamW con learning_rate
SINO SI optimizer ES "adamax":
    ASIGNAR optimizer como Adamax con learning_rate
SINO SI optimizer ES "nadam":
    ASIGNAR optimizer como Nadam con learning_rate
SINO SI optimizer ES "adam":
    ASIGNAR optimizer como Adam con learning_rate
SINO:
    ASIGNAR optimizer como RMSprop con learning_rate

COMPILAR modelo CON:
    optimizador = optimizer
    pérdida = "mean_squared_error"
    métricas = ["mae", "rmse", "r2_score"] #Cálculo de otras métricas
```

Listing 8.9: definición de los optimizadores y las métricas de evaluación utilizadas en la búsqueda bayesiana.

8.3 Entrenamiento y evaluación del modelo RNN

Para el entrenamiento de modelos predictivos de redes neuronales, el uso de un entorno con GPU es esencial debido a su alta capacidad de procesamiento en paralelo y su gran ancho de banda de memoria. A diferencia de las CPU, las GPUs pueden manejar eficientemente los grandes volúmenes de parámetros, activaciones y gradientes que se actualizan en cada paso del entrenamiento, lo que acelera significativamente el proceso [46].

Dado el límite de recursos disponibles, es esencial aplicar técnicas de regularización y control de entrenamiento para maximizar la eficiencia y minimizar los costos computacionales durante la búsqueda de hiperparámetros.

Determinar el número óptimo de épocas para alcanzar la mínima pérdida de validación puede resultar desafiante. Un método tradicional, aunque ineficiente, es entrenar el modelo hasta observar que el aprendizaje se estabiliza, identificar el punto ideal y luego reiniciar el entrenamiento. Sin embargo, una alternativa más efectiva es utilizar *Early Stopping*, que detiene el entrenamiento cuando la pérdida de validación deja de mejorar, evitando el sobreajuste y reduciendo el tiempo de cómputo. Asimismo, el uso de *ReduceLROnPlateau* permite ajustar dinámicamente la tasa de aprendizaje cuando la pérdida

se estabiliza, previniendo el estancamiento en mínimos locales y mejorando la eficiencia [51]. En el listing 8.10 se puede observar la aplicación de ambos métodos durante la búsqueda bayesiana.

- *EarlyStopping*: mediante la definición de un límite de paciencia, detiene el entrenamiento con la combinación actual cuando la pérdida de validación no mejora después del número determinado de épocas definido.
- *ReduceLROnPlateau*: mediante la definición de un límite de paciencia, reduce la tasa de aprendizaje si la pérdida de validación no mejora después de un número determinado de épocas. Tras lo cual reduce la tasa de aprendizaje por un factor.

```
DEFINIR early_stopping COMO UNA CONFIGURACIÓN QUE:
    MONITOREA "val_loss"
    TIENE paciencia = 15
    RESTAURA LOS MEJORES PESOS CUANDO SEA NECESARIO

DEFINIR lr_scheduler COMO UNA CONFIGURACIÓN QUE:
    MONITOREA "val_loss"
    USA UN FACTOR DE REDUCCIÓN = 0.3
    TIENE paciencia = 4
    ESTABLECE UNA TASA DE APRENDIZAJE MÍNIMA = 1e-6
```

Listing 8.10: definición de los parámetros de regularización y optimización durante las pruebas.

Tras evaluar distintas combinaciones de estructuras e iterar con hiperparámetros, hemos obtenido la mejor pérdida de validación de 0,3320 tras 8 épocas de entrenamiento y los hiperparámetros mostrados en el listing 8.11.

```
"units_lstm_encoder": 100,
"activation_encoder": "relu",
"l2_regularization_encoder": 0.00002,
"dropout_lstm": 0.2,
"num_heads_attention": 4,
"key_dim_attention": 32,
"dropout_attention": 0.4,
"dense_units": 300,
"l2_dense": 0.00004,
"activation_dense": 'elu',
"dropout_rate": 0.2,
"optimizer": "adam",
"learning_rate": 0.00005
```

Listing 8.11: hiperparámetros óptimos identificados.

La figura 8.4 muestra que el modelo de redes neuronales basado en **GRU** entrena de manera efectiva. Se puede observar como el modelo aprende de forma muy eficiente en las primeras 12 épocas, tanto entrenamiento como validación mejoran de manera coordinada para las métricas, como *MSE*, *MAE* y *RMSE*, junto con un aumento en el *R²*, lo que indica que el modelo está capturando patrones relevantes y ajustándose correctamente a los datos. Tras un período inicial de aprendizaje acelerado durante las primeras tres métricas, la velocidad de mejora disminuye progresivamente, reduciendo la efectividad del aprendizaje.

No obstante, el modelo se continúa refinando de manera gradual, donde la pérdida mejora cada vez menos, y finalmente el *lr_scheduler*, configurado para disminuirla en un 20% tras dos épocas consecutivas sin mejora en la pérdida de validación, actúa en la época 13

y en adelante para intentar exprimir un mínimo ajuste, pero el modelo no lo consigue, alcanzando un punto de estabilización, donde las métricas tanto de validación como de entrenamiento se mantienen constantes.

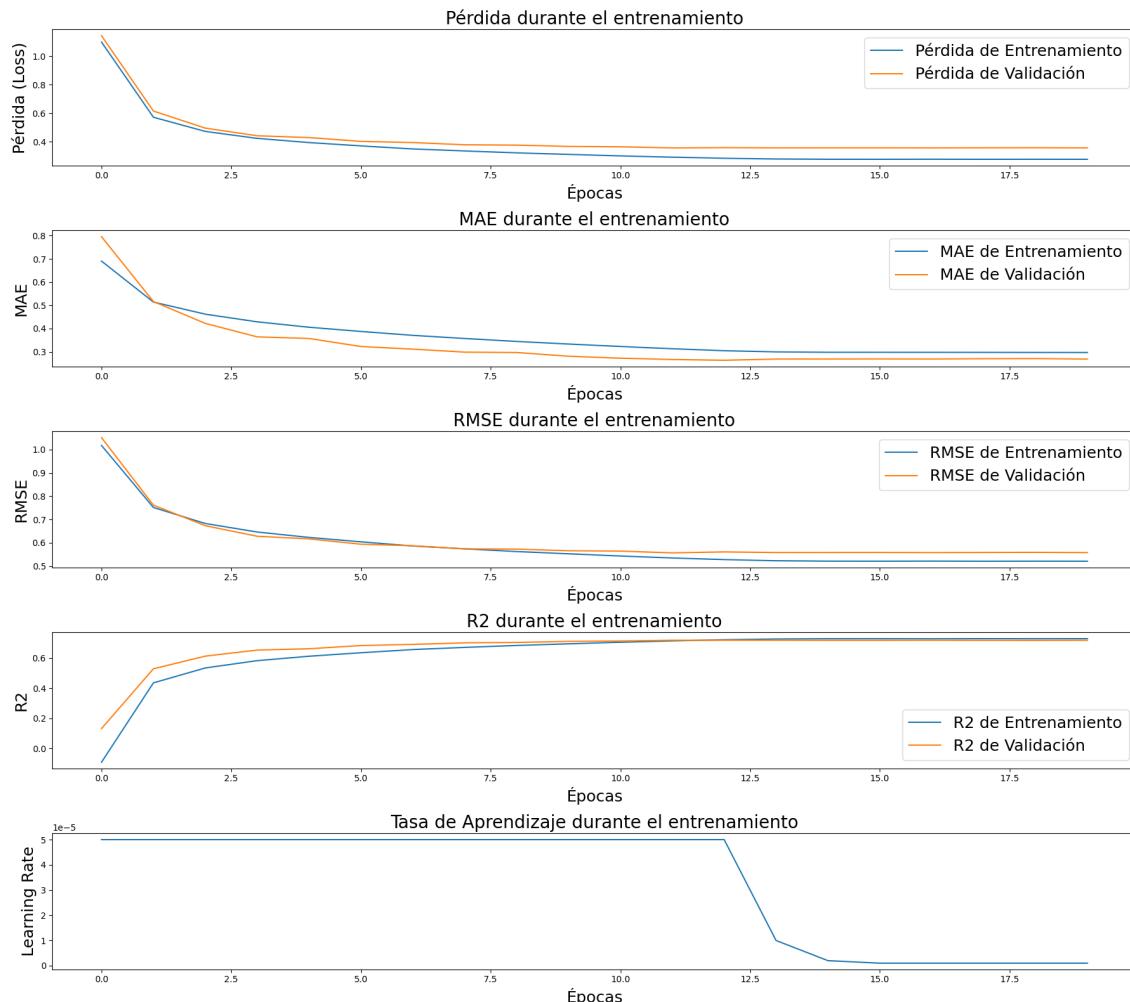


Figura 8.4: métricas evaluación durante el entrenamiento del modelo de redes neuronales.

La disminución constante de la pérdida hasta llegar a la estabilización refleja una mejora continua en la capacidad predictiva del modelo. De manera similar, la reducción progresiva del MAE y el RMSE indica una disminución gradual de los errores.

El aumento constante del R^2 sugiere que el modelo es cada vez más capaz de explicar la variabilidad de los datos. Además, la pequeña diferencia entre las métricas de entrenamiento, que siguen una tendencia similar y se estabilizan juntas, señala una generalización adecuada del modelo.

Comprobamos en la figura 8.5 la predicción de los primeros 168 períodos y visualizamos la comparación con los valores reales. Se observa que el modelo logra capturar correctamente la tendencia general, ajustándose al patrón de transacciones de la plataforma de nuestro cliente. No obstante, también se aprecia que, en determinados casos, las predicciones tienden a subestimar los valores reales de manera recurrente, lo que sugiere cierta dificultad del modelo para adaptarse con precisión a algunas fluctuaciones específicas.

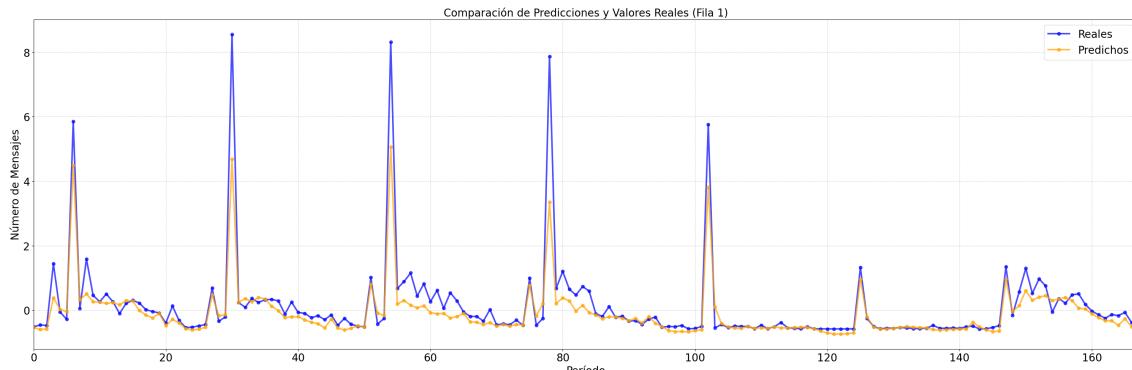


Figura 8.5: predicciones del modelo de redes neuronales durante los primeros 168 períodos.

Un entrenamiento prolongado puede llevar al sobreajuste, donde el modelo memoriza detalles y ruido presentes en los datos de entrenamiento, afectando negativamente su rendimiento al enfrentarse a datos no vistos. Por ello, es esencial monitorear su desempeño en conjuntos de validación y prueba, ya que esto permite determinar el momento óptimo para finalizar el entrenamiento y asegurar una adecuada capacidad de generalización [65]. Por ello, una vez entrenado el modelo, procedemos a combinar los conjuntos de entrenamiento y validación para reentrenarlo y aplicarlo a un conjunto de prueba previamente no utilizado.

Una vez entrenado el modelo, es fundamental evaluar no solo su rendimiento inmediato, sino también su capacidad para realizar predicciones a largo plazo en diferentes condiciones. Para ello, se han empleado dos enfoques, uno iterativo y otro recursivo, tal y como se muestra en el listing 8.12 que permiten analizar tanto la estabilidad como la fiabilidad de las predicciones.

```
# Predicción con entradas reales (Iterativa)
INICIALIZAR lista_predicciones_iterativa COMO lista vacía

PARA i DESDE 0 HASTA (longitud de X_test - 168 + 1) CON PASO 168 HACER:
  DEFINIR X_ventana COMO subconjunto de X_test DESDE i HASTA i + 168
  PREDECIR y_pred_ventana USANDO model SOBRE X_ventana
  AÑADIR y_pred_ventana A lista_predicciones_iterativa

CONCATENAR lista_predicciones_iterativa EN y_pred_real_input

# Predicción recursiva (Autónoma)
INICIALIZAR lista_predicciones_recursiva COMO lista vacía
DEFINIR X_input COMO la primera ventana real DE X_test DE dimensión (1, 168, n_features)
CALCULAR num_blocks COMO el número de bloques completos de 168 en X_test

PARA cada bloque i DESDE 0 HASTA num_blocks - 1 HACER:
  PREDECIR y_pred_ventana USANDO model SOBRE X_input
  AÑADIR y_pred_ventana[0] A lista_predicciones_recursiva

  DESPLAZAR X_input 168 posiciones HACIA ATRÁS EN EL EJE TEMPORAL
  SUSTITUIR últimos 168 valores de la variable objetivo en X_input
  CON las predicciones y_pred_ventana

CONVERTIR lista_predicciones_recursiva A ARRAY CON FORMA (-1, 168, 1)

# Preparación de resultados para análisis
REMODELAR y_pred_real_input, lista_predicciones_recursiva Y y_test A VECTORES PLANOS
AJUSTAR todas las listas al mínimo tamaño común

CREAR DataFrame df_predictions CON COLUMNAS:
  'Reales', 'Predicciones_iterativas', 'Predicciones_recursivas'
```

Listing 8.12: proceso de predicción iterativa y recursiva con el modelo de redes neuronales.

El primer enfoque consiste en predecir de manera iterativa ventanas de 168 períodos, utilizando siempre datos reales como entrada. Esta estrategia permite evaluar cómo de bien es capaz el modelo de anticipar futuros valores cuando dispone en cada paso de información precisa y conocida, reflejando su ajuste directo sobre datos fiables. En la segunda gráfica de la figura 8.6 puede observarse el contraste entre los valores reales y las predicciones, que tienden a ser conservadoras. Cuando el modelo no detecta señales claras en los datos de entrada, opta por mantenerse dentro de un rango seguro, aproximándose a un valor promedio en lugar de seguir la tendencia descendente que muestran los valores reales. Este comportamiento es habitual en modelos que buscan minimizar el error global, priorizando la estabilidad sobre la sensibilidad a cambios locales. No obstante, a pesar de su naturaleza conservadora, las predicciones logran capturar de forma razonable tanto la tendencia general como las fluctuaciones en el número de mensajes transaccionados.

El segundo enfoque evalúa la capacidad del modelo para generar predicciones de forma recursiva, comenzando con una semana de datos reales y utilizando, a partir de ese momento, sus propias predicciones como entrada para las siguientes iteraciones, en lugar de emplear datos reales. Esta aproximación es especialmente útil para comprobar si el modelo mantiene coherencia y estabilidad en escenarios más cercanos a un entorno de producción, donde las futuras predicciones dependen directamente de sus propias estimaciones previas, sin intervención de datos externos. En la primera gráfica de la figura 8.6 se ilustran los valores reales junto con las predicciones correspondientes a las primeras cinco ventanas temporales generadas con este método, para las cuales puede observarse cómo el error tiende a acumularse progresivamente, provocando un desfase creciente que afecta a la precisión de pronósticos.

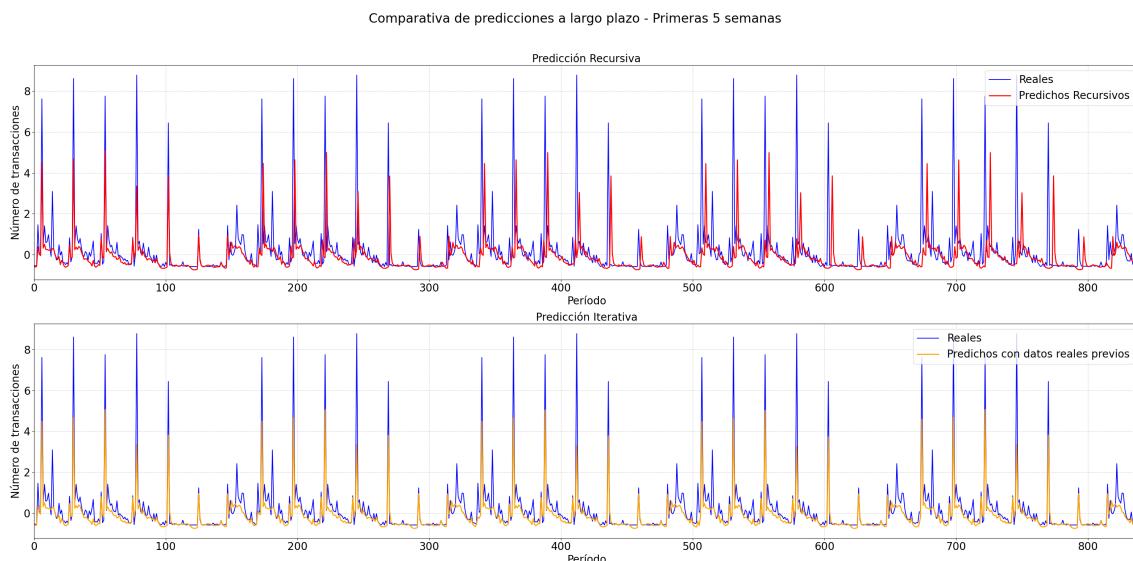


Figura 8.6: comparación entre los valores reales y las predicciones del número de mensajes transaccionados, utilizando enfoques iterativo y recursivo a partir del modelo de red neuronal.

Comparar ambos métodos permite valorar la robustez del modelo tanto en entornos controlados como en simulaciones de despliegue real, garantizando que su rendimiento no dependa exclusivamente de condiciones ideales y que sea capaz de sostener predicciones estables en cadenas prolongadas sin una degradación significativa. En la figura 8.7 se muestran las medias por período temporal de las ventanas de 168 períodos, junto con los valores reales y las predicciones obtenidas mediante ambos enfoques. Puede observarse

que las predicciones generadas de forma recursiva, al acumular un desfase progresivo, presentan una diferencia notable respecto a los valores reales. En contraste, las predicciones iterativas, que emplean datos reales como entrada, logran adaptarse de manera mucho más precisa a la tendencia, aunque tienden a subestimar ligeramente los valores reales.

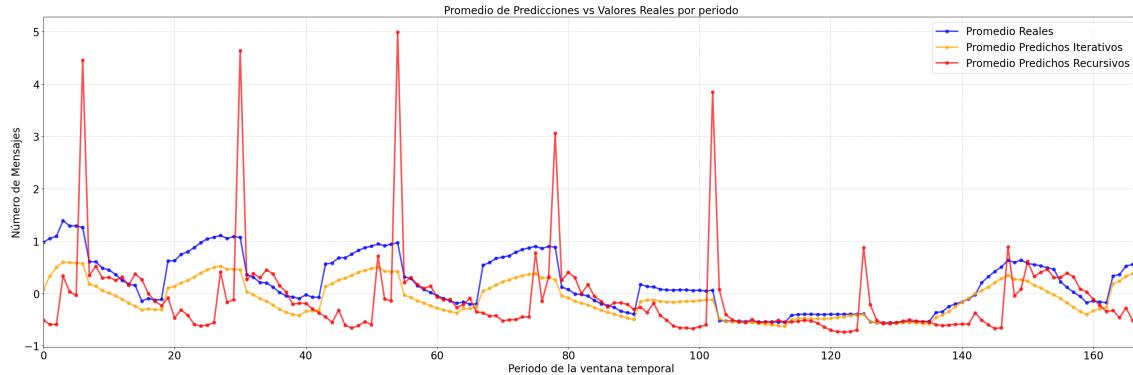


Figura 8.7: comparación media por período entre los valores reales y las predicciones del número de mensajes transaccionados, utilizando enfoques iterativo y recursivo a partir del modelo de red neuronal.

Cabe destacar que, en los últimos 90 períodos semanales, las predicciones iterativas muestran una convergencia clara hacia los valores reales, lo que sugiere que el modelo ha alcanzado un punto de equilibrio en su aprendizaje, siendo capaz de replicar con mayor fidelidad la dinámica de la variable a lo largo del tiempo.

En definitiva, el modelo de predicción de series temporales de redes neuronales recurrentes, ha demostrado ser una herramienta válida para anticipar tendencias y capturar la dinámica global del número de mensajes transaccionados, aunque con ciertas limitaciones en cuanto a la precisión de los valores estimados y la profundidad de la predicción autónoma. Los resultados evidencian que, cuando dispone de datos reales en cada ventana, el modelo es capaz anticipar la tendencia general de forma estable, aunque subestimando los valores reales. Sin embargo, bajo un enfoque recursivo, donde las propias predicciones alimentan las entradas futuras, el error se acumula progresivamente, limitando la fiabilidad de las estimaciones más allá de la segunda ventana temporal.

CAPÍTULO 9

Conclusiones y resultados

Este proyecto ha demostrado el valor de aplicar modelos de segmentación y predicción a datos de monitorización de flujos transaccionales EDI, con el fin de dotar a Edicom de herramientas avanzadas para anticipar y comprender patrones de comportamiento para tratar de optimizar tanto la gestión operativa como la toma de decisiones estratégicas.

En primer lugar, el modelo de predicción basado en redes neuronales recurrentes ha evidenciado su capacidad para capturar la dinámica general del tráfico de mensajes intercambiados, aportando estimaciones útiles para la empresa. A pesar de que el modelo dependa de sus propias predicciones para pronosticar horizontes temporales largos, lo que puede provocar una acumulación de errores, los resultados obtenidos constituyen una base sólida sobre la cual continuar desarrollando sistemas predictivos robustos. Al poder predecir la tendencia y las fluctuaciones del tránsito de mensajes, es posible anticiparse a picos de demanda, mejorar la distribución de recursos y evaluar el impacto de posibles cambios en la plataforma, reforzando así la estrategia de planificación de Edicom.

En el ámbito de la segmentación, la aplicación del algoritmo K-Means sobre un espacio reducido mediante análisis de componentes principales ha permitido agrupar intercambios en función de sus características más relevantes, como son frecuencia, tamaño, tipo de mensaje, origen, destino, configuración de seguridad, y variables temporales, entre otras derivadas. La segmentación resultante ha mostrado coherencia y distinción de dinámicas de transmisiones por hora, logrando identificar grupos significativos que reflejan dinámicas de uso diferenciados. Este enfoque puede facilitar el diseño de estrategias de personalización de servicios para el cliente, optimización de modelos de facturación y permitir un análisis más ágil para clientes con entornos transaccionales más complejos.

Una vez obtenidos los resultados, estos se han incorporado en la herramienta de almacenamiento de Edicom, LTA, siguiendo el procedimiento descrito en el apartado 11.3 de los anexos. Una vez integrados, conforme a lo indicado en la sección 11.4, los datos pueden ser representados gráficamente de forma didáctica mediante la aplicación de Edicom Analytics. Esta funcionalidad contribuye a presentar los resultados obtenidos, facilitando un mejor traspaso de conocimientos y facilitando la comprensión de nuestros clientes, permitiéndonos identificar sus patrones y comportamientos.

La presentación de los resultados se ha abordado desde dos enfoques complementarios. En primer lugar, la visualización de los resultados del modelo de clusterización **K-Means** según el tamaño de las transacciones (figura 11.12) y el volumen de tráfico de mensajes (figura 11.11). Estos dos dashboards han permitido visualizar la segmentación de los patrones operacionales horarios de los clientes a partir de las métricas más representativas extraídas de los datos de monitorización. En segundo lugar, el dashboard generado a partir de los resultados de la predicción del número de mensajes intercambiados mediante redes neuronales recurrentes (figura 11.10) permite analizar las diferencias entre los valores reales y los estimados, evaluar la precisión de las predicciones futuras y estimar su posible impacto en la toma de decisiones estratégicas de la empresa.

En resumen, la evaluación de los tres dashboards desarrollados han demostrado su capacidad para revelar información valiosa no perceptible a simple vista. Estos resultados representan una aportación significativa tanto para la mejora de los procesos internos de la empresa como para la toma de decisiones estratégicas en futuros proyectos. Todo ello

refuerza el valor de integrar herramientas de analítica avanzada en la gestión operativa y comercial.

CAPÍTULO 10

Relación del trabajo desarrollado con los estudios cursados y trabajos futuros

Durante el desarrollo del proyecto, adaptar y escalar los modelos a un caso práctico real supuso un desafío significativo. Las tareas de limpieza, modelado, validación e interpretación de los datos exigieron una profundización en áreas previamente abordadas durante el Grado en Ciencia de Datos, especialmente en análisis avanzado, ingeniería de variables y ajuste de modelos de clustering y predicción.

Esta experiencia ha representado un proceso de aprendizaje continuo, permitiéndome consolidar conocimientos, adquirir nuevas habilidades y comprender de forma práctica cómo los datos pueden orientar decisiones estratégicas dentro de una organización. La aplicación de los conocimientos adquiridos sobre datos de monitorización de transacciones EDI refleja claramente la conexión entre la formación académica y las necesidades reales de análisis en entornos empresariales.

Además, la integración de los resultados en una herramienta de visualización ha permitido maximizar el valor de los datos y detectar oportunidades clave, respaldando una toma de decisiones informada y eficiente. En este sentido, la preparación en técnicas de interpretación y presentación de resultados ha sido esencial para comunicar los hallazgos de forma clara y comprensible.

Los resultados obtenidos constituyen una base sólida para futuras mejoras y ampliaciones, como la incorporación de nuevas fuentes de datos, la optimización continua de los modelos y la posible implementación de sistemas de alerta temprana. Todo ello contribuye a fortalecer el uso de la analítica avanzada y la inteligencia de negocio dentro de la organización. A partir de este trabajo, se han identificado diversas líneas de evolución que podrían enriquecer tanto la profundidad como la aplicabilidad de los resultados en contextos reales. Entre ellas destacan las siguientes:

- **Automatización del flujo de datos:** una de las principales mejoras pendientes es, mediante la conexión directa con las herramientas y sistemas de información de la empresa, la automatización integral del proceso de transformación e integración de los datos procedentes de las trazas transaccionales. Este avance permitiría actualizar los datos de forma continua y eficiente, además de facilitar la recalibración periódica de los modelos, asegurando así su precisión y capacidad de adaptación ante posibles cambios en el tráfico y perfil del cliente.
- **Predicción de otras métricas:** la incorporación de predicciones sobre otras variables de monitorización relevantes para las transacciones, como el tamaño, el volumen específico del flujo entrante o saliente, o el volumen entre interlocutores concretos, permitiría ampliar significativamente el conocimiento sobre el comportamiento del cliente. Esta información enriquecería el análisis y aportaría una visión más completa para la toma de decisiones estratégicas.

- **Analítica avanzada a partir de otras fuentes:** la integración de datos procedentes de otros departamentos, en particular del área de soporte, donde se gestionan tareas, solicitudes e incidencias, junto con la información de la ficha del cliente, permitirá enriquecer y ampliar el análisis hacia una visión más global del cliente y sus interacciones. La aplicación de analítica avanzada sobre estos nuevos datos abriría nuevas oportunidades para una comprensión más precisa y personalizada del perfil de los clientes, fortaleciendo así las estrategias de atención y fidelización.
- **Exploración de otras técnicas avanzadas analíticas:** la exploración e incorporación de técnicas de inteligencia artificial más avanzadas para las tareas de clasificación, segmentación y predicción representa un camino prometedor. Probar otros enfoques y modelos permitiría aumentar la precisión de los resultados, mejorar la eficiencia de los modelos actuales y, en última instancia, incrementar el valor estratégico de los análisis realizados.

Por último, es importante destacar que, durante el desarrollo del trabajo, también se identificaron enfoques que, si bien resultaban atractivos en un primer momento, demostraron ser poco recomendables. En concreto, se observó que la inclusión de variables irrelevantes o mal depuradas introduce ruido en los modelos de redes neuronales, perjudicando tanto su rendimiento como su interpretabilidad. Por este motivo, se concluye que la selección rigurosa de variables combinando criterios estadísticos y conocimiento experto sobre el flujo EDI del cliente, es un aspecto clave para asegurar la calidad y la fiabilidad de los resultados.

Bibliografía

- [1] EDICOM. *Conozca la tecnología que hay detrás de nuestra plataforma EDI.* s.l.: Edicom, 2022. [en línea]. Disponible en: <https://edicomgroup.es/blog/tecnologia-del-software-edi-de-edicom> [Consulta: 03/06/2024]
- [2] EDICOM. *Integración de aplicaciones iPaaS - Integration Platform As A Service.* s.l.: Edicom, 2025. [en línea]. Disponible en: <https://edicomgroup.es/ipaas> [Consulta: 03/06/2024]
- [3] EDICOM. *Conecte, visualice, comparta.* Edicom, 2025. [en línea]. Disponible en: <https://edicomgroup.es/analytics-services> [Consulta: 03/06/2024]
- [4] Gupta, Ankit. *Electronic Data Interchange (EDI) Software Market Research.* s.l.: Market research future, 2025. [En línea]. Disponible en: <https://www.marketresearchfuture.com/reports/electronic-data-interchange-edi-software-market-11537> [Consulta: 03/06/2024]
- [5] The Insight Partners. *Electronic Data Interchange (EDI) Market worth \$58.98 Billion by 2030 - Exclusive Report by The Insight Partners.* s.l: Globe Newswire, 2023. [En línea]. Disponible en: <https://www.globenewswire.com/news-release/2023/09/21/2747427/0/en/Electronic-Data-Interchange-EDI-Market> [Consulta: 15/06/2024]
- [6] The Business Research Company. *Electronic Data Interchange (EDI) Software Global Market Report 2025.* s.l.: The Business Research Company, 2025. [En línea]. Disponible en: <https://www.thebusinessresearchcompany.com/report/electronic-data-interchange-edi-software-global-market-report> [Consulta: 03/01/2025]
- [7] Astera. *EDIConnect: Solución de intercambio electrónico de datos (EDI) de nivel empresarial.* s.l.: Astera, 2025. [En línea]. Disponible en: <https://www.astera.com/es/solutions/technology-solutions/edi-solution/> [Consulta: 03/06/2024]
- [8] IBM. *Soluciones de intercambio electrónico de datos (EDI) y API B2B.* s.l.: IBM, 2024. [En línea]. Disponible en: <https://www.ibm.com/es-es/electronic-data-interchange> [Consulta: 03/06/2024]
- [9] IBM. *IBM Sterling B2B Integration Suite.* s.l: IBM, 2024. [En línea]. Disponible en: <https://www.ibm.com/downloads/documents/us-en/10a99803c7afabd> [Consulta: 03/06/2024]
- [10] OpenText. *OpenText Business Network Cloud.* s.l.: OpenText, 2025. [En línea]. Disponible en: <https://www.opentext.com/es-es/productos/business-network-cloud> [Consulta: 03/06/2024]
- [11] OpenText. *OpenText Trading Grid Command Center.* s.l.: OpenText, 2025. [En línea]. Disponible en: <https://www.opentext.com/es-es/productos/trading-grid-command-center> [Consulta: 03/06/2024]
- [12] Cleo. *Visibility & Business Insights.* s.l.: Cleo, 2025. [En línea]. Disponible en: <https://www.cleo.com/solutions/use-case/visibility-business-insights> [Consulta: 03/06/2024]

- [13] Cleo. *Cleo Integration Cloud*. s.l.: Cleo, 2025. [En línea]. Disponible en: <https://www.cleo.com/cleo-integration-cloud> [Consulta: 03/06/2024]
- [14] Mitek Systems. *No todo son ciberataques: los 5 principales riesgos del Big Data*. s.l.: Mitek Systems, 2023. [En línea]. Disponible en: <https://www.miteksystems.com/es/blog/5-principales-riesgos-big-data> [Consulta: 20/08/2024]
- [15] Universidad Francisco de Vitoria. *¿Cuáles son las principales desventajas del Big Data?*. Vitoria: Universidad Francisco de Vitoria, 2023. [En línea]. Disponible en: <https://www.ufv.es/principales-desventajas-big-data> [Consulta: 03/01/2025]
- [16] Burke, Tim. *Harnessing the Power of Log Analytics for Your Business*. s.l.: Quest Technology Management, 2023. [En línea]. Disponible en: <https://questsys.com/ceo-blog/harnessing-the-power-of-log-analytics-for-your-business/> [Consulta: 20/08/2024]
- [17] OpsWorks. *Log Data Analysis: Why Is It Important?*. s.l.: OpsWorks, 2022. [En línea]. Disponible en: <https://www.opsworks.co/blog/log-data-analysis-why-is-it-important> [Consulta: 20/08/2024]
- [18] AEPD. *Anonymisation and pseudonymisation*. s.l.: AEPD (Agencia Española de Protección de Datos), 2021. [En línea]. Disponible en: <https://www.aepd.es/en/prensa-y-comunicacion/blog/anonymisation-and-pseudonymisation> [Consulta: 23/08/2024]
- [19] AEPD. *Orientaciones y garantías en los procedimientos de ANONIMIZACIÓN de datos personales*. s.l.: AEPD (Agencia Española de Protección de Datos), 2021. [En línea]. Disponible en: <https://www.aepd.es/guias/guia-orientaciones-procedimientos-anonimizacion.pdf> [Consulta: 23/08/2024]
- [20] Autoridad Nacional de Protección de Datos de Singapur. *Guía básica de anonimización*. s.l.: AEPD (Agencia Española de Protección de Datos), 2022. [En línea]. Disponible en: <https://www.aepd.es/documento/guia-basica-anonimizacion.pdf> [Consulta: 23/08/2024]
- [21] InnovacionTech. *Éxito en proyectos de Big Data: Claves y métricas efectivas*. s.l.: InnovacionTech, 2025. [En línea]. Disponible en: <https://innovaciontech.com/exito-en-proyectos-de-big-data-claves-y-metricas-efectivas/> [Consulta: 23/08/2024]
- [22] Data Universe. *Evaluación ética de modelos predictivos en ciencia de datos*. s.l.: Data Universe, 2025. [En línea]. <https://data-universe.org/evaluacion-etica-de-modelos-predictivos-en-ciencia-de-datos/> [Consulta: 10/02/2025]
- [23] Keyrus. *Las 11 técnicas más utilizadas en el modelado de análisis predictivo*. s.l.: Keyrus, 2025. [En línea]. Disponible en: <https://keyrus.com/sp/es/insights/las-11-tecnicas-mas-utilizadas-en-el-modelado-de-analisis-predictivos> [Consulta: 10/02/2025]
- [24] Pérez, Anna. *Riesgos en proyectos de Big Data*. s.l.: OBS Business School, 2016. [En línea]. Disponible en: <https://www.obsbusiness.school/blog/riesgos-en-proyectos-de-big-data> [Consulta: 12/12/2024]

- [25] Faster Capital. *Mejora de los procesos de análisis de riesgos*. s.l.: Faster Capital, 2024. [En línea]. Disponible en: <https://fastercapital.com/es/tema/mejora-de-los-procesos-de-an%C3%A1lisis-de-riesgos.html> [Consulta: 06/01/2025]
- [26] Manyika, James; Chui, Michael; Brown, Brad; Bughin, Jacques; Dobbs, Richard; Roxburgh, Charles; Hung Byers, Angela. *Big Data: The Next Frontier for Innovation, Competition and Productivity*. s.l.: McKinsey Global Institute, 2011. [En línea]. Disponible en: <https://es.slideshare.net/slideshow/big-data-innovation> [Consulta: 03/03/2025]
- [27] Cam Gensollen, César Rogelio. *Big data en el mundo del retail, segmentación de clientes y sistema de recomendación en una cadena de supermercados de Europa*. s.l.: Research Gate, 2022. [En línea]. Disponible en: https://www.researchgate.net/publication/360279021_Big_Data/ [Consulta: 10/02/2025]
- [28] Grupo Hasten. *Big Data y análisis predictivo en la toma de decisiones empresariales*. s.l.: Hasten Group, 2025. [En línea]. Disponible en: <https://www.grupohasten.com/big-data-y-analisis-predictivo-en-la-toma-de-decisiones-empresariales/> [Consulta: 10/02/2025]
- [29] Vercheval, Sarah. *Inteligencia artificial y Big Data: cómo funcionan, diferencias y su relación*. s.l.: InboundCycle, 2024. [En línea]. Disponible en: <https://www.inboundcycle.com/blog-de-inbound-marketing/inteligencia-artificial-y-big-data> [Consulta: 05/10/2024]
- [30] Gil, Elena. *Big data, privacidad y protección de datos*. s.l.: AEPD (Agencia Española de Protección de Datos), 2015. [En línea]. Disponible en: <https://www.aepd.es/sites/default/files/2019-10/big-data.pdf> [Consulta: 03/06/2024]
- [31] Gordon, Joshua. *Practical Guide for Feature Engineering of Time Series Data*. s.l.: Dot Data, 2024. [En línea]. Disponible en: <https://dotdata.com/blog/practical-guide-for-feature-engineering-of-time-series-data/> [Consulta: 25/11/2024]
- [32] Han, Jiawei; Kamber, Micheline; Pei, Jian. *Data Mining: Concepts and Techniques*. s.l.: Morgan Kaufmann, 2011. [En línea]. Disponible en: <https://www.lecturus.org/wp-content/uploads/2024/07/Data-Mining-Concepts-and-Techniques-3rd-ed.-Jiawei-Han-Micheline-Kamber-Jian-Pei.pdf> [Consulta: 23/08/2024]
- [33] Hua, Yuxiu; Zhao, Zhifeng; Li, Rongpeng; Chen, Xianfu; Liu, Zhiming; Zhang, Honggang. *Deep Learning with Long Short-Term Memory for Time Series Prediction*. s.l.: IEEE (Institute of Electrical and Electronic Engineers) Communications Magazine, 2019. [En línea] Disponible en: <https://ieeexplore.ieee.org/document/8663965> [Consulta: 23/08/2024]
- [34] Hochreiter, Sepp; Schmidhuber, Jürgen. *Long Short-Term Memory*. s.l.: MIT Press Neural Computation, 1997. [En línea] Disponible en: <https://ieeexplore.ieee.org/abstract/document/6795963> [Consulta: 23/08/2024]
- [35] Cho, Kyunghyun; van Merriënboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger; Bengio, Yoshua. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, Doha (Qatar): Cornell University, Association for Computational Linguistics, 2014. [Consulta: 23/08/2024]

- [36] Saini, Komal; Sharma, Sandeep. *Gated Recurrent Unit (GRU) in RNN for traffic forecasting based on time-series data*. Dehradun, India, 2nd International CISCT (Conference on Innovative Sustainable Computational Technologies): IEEE (Institute of Electrical and Electronic Engineers) Communications Magazine Communications Magazine, 2023. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/10046484> [Consulta: 23/08/2024]
- [37] Komal Saini; Sandeep Sharma. *Gated Recurrent Unit (GRU) in RNN for Traffic Forecasting Based on Time-Series Data*. s.l.: IEEE (Institute of Electrical and Electronic Engineers) Communications Magazine Communications Magazine, 2022. Disponible en: <https://ieeexplore.ieee.org/document/10046484> [Consulta: 13/09/2024]
- [38] Chauhan, Nagesh Singh. *Optimización de hiperparámetros para modelos de aprendizaje automático*. s.l.: DataSource AI, 2020. [En línea]. Disponible en: <https://www.datasource.ai/es/data-science-articles/optimizacion-de-hiper-parametros-para-modelos-de-aprendizaje-automatico> [Consulta: 13/09/2024]
- [39] Lopez, Diego. *Estadística para Ciencia de Datos: Una Guía Completa para Aspirantes a Practicantes de ML*. s.l.: FreecodeCamp, 2024. [En línea]. Disponible en: <https://www.freecodecamp.org/espanol/news/estadistica-para-ciencia-de-datos> [Consulta: 22/06/2024]
- [40] Casas, Pablo. *Libro Vivo de Ciencia de Datos*. s.l.: Escuela de Datos Vivos, 2019. [En línea]. Disponible en: <https://librovivodecienciadedatos.ai/preparacion-de-datos.html> [Consulta: 22/06/2024]
- [41] Kis, András. *Understanding Autocorrelation and Partial Autocorrelation Functions (ACF and PACF)*. s.l.: Medium, 2024. [En línea]. Disponible en: <https://medium.com/understanding-autocorrelation-and-partial-autocorrelation> [Consulta: 12/12/2024]
- [42] DataCalculus. *Manejo de datos de alta dimensionalidad en análisis: técnicas y estrategias*. s.l.: DataCalculus, 2025. [En línea]. Disponible en: <https://datacalculus.com/es/centro-de-conocimiento/anal%C3%ADtica-de-datos/an%C3%A1lisis-de-datos/anejando-datos-de-alta-dimensionalidad-en-analisis> [Consulta: 03/03/2025]
- [43] Chung, Junyoung; Gulcehre, Caglar; Cho, KyungHyun; Bengio, Yoshua. *Empirica Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. s.l.: Cornell University, 2014. [En línea]. Disponible en: <https://arxiv.org/abs/1412.3555> [Consulta: 19/03/2024]
- [44] Jorge-Botana, Guillermo. *Redes neuronales recurrentes y Transformers para modelos cognitivos del lenguaje*. s.l.: ResearchGate, 2024. Disponible en: https://www.researchgate.net/publication/382073083_Redes_neuronales_recurrentes_y_Transformers_para_modelos_cognitivos_del_lenguaje [Consulta: 19/03/2024]
- [45] Fernández, Salguero. *Series Temporales Avanzadas: Aplicación de Redes Neuronales para el Pronóstico de Series de Tiempo*. s.l.: Universidad de Granada, 2021. [En línea]. Disponible en: https://masteres.ugr.es/estadistica-aplicada/sites/master/moea/public/inline-files/TFM_FernAndez%20SalgueroRicardo%20Alonzo.pdf [Consulta: 19/03/2024]
- [46] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. *Deep Learning*. s.l: MIT Press, 2016. [En línea]. Disponible en: <https://www.deeplearningbook.org/> [Consulta: 20/03/2024]

- [47] Gordon, Joshua. *Practical Guide for Feature Engineering of Time Series Data*. s.l.: DotData, 2025. [En línea]. Disponible en: <https://dotdata.com/blog/practical-guide-for-feature-engineering-of-time-series-data/> [Consulta: 20/03/2024]
- [48] Vidip, Jain. *Understanding Train, Test, and Validation Data in Machine Learning*. s.l.: Medium, 2024. [En línea]. Disponible en: <https://medium.com/@jainvidip/understanding-train-test-validation-data> [Consulta: 22/03/2024]
- [49] Amat Rodrigo, Joaquín. *Redes neuronales con Python*. s.l.: Ciencia de Datos, 2021. [En línea]. Disponible en: <https://cienciadedatos.net/documentos/py35-redes-neuronales-python> [Consulta: 20/03/2024]
- [50] Mucci, Tim. *What is data leakage in machine learning?*. s.l.: IBM, 2024. [En línea]. Disponible en: <https://www.ibm.com/think/topics/data-leakage-machine-learning> [Consulta: 22/03/2024]
- [51] Chollet, François. *Deep Learning with Python*. Shelter Island, NY: Manning, 2018. [En línea]. Disponible en: <https://github.com/anishLearnstoCode/books/blob/master/machine-learning/deep-learning-with-python-francois-chollet.pdf> [Consulta: 20/03/2024]
- [52] Jiang, Feng; Liu, Jie; Yi, Chunzhi. *Comparative Analysis of the Clustering Quality in Self-Organizing Maps for Human Posture Classification*. s.l: NCBI (National Center for Biotechnology Information), 2023. [En línea]. Disponible en: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10538130/#sec6-sensors-23-07925> [Consulta: 18/03/2024]
- [53] LaViale, Trevor. *Understanding HDBSCAN: A Deep Dive into Hierarchical Density-Based Clustering*. s.l.: Arize AI, 2025. [En línea]. Disponible en: <https://arize.com/blog-course/understanding-hierarchical-density-based-clustering> [Consulta: 10/02/2025]
- [54] Zhang, Pengfei; Xue, Jianru; Lan, Cuiling; Zeng, Wenjun; Gao, Zhanning; Zheng, Nanning. *Adding Attentiveness to the Neurons in Recurrent Neural Networks*. s.l.: Cornell University, 2018. [En línea]. Disponible en: <https://arxiv.org/abs/1807.04445> [Consulta: 22/03/2025]
- [55] Dudek, Grzegorz; Smyl, Sławek; Pełka, Paweł. *Recurrent Neural Networks for Forecasting Time Series with Multiple Seasonality: A Comparative Study*. s.l.: Cornell University, 2022. [En línea]. Disponible en: <https://arxiv.org/abs/2203.09170> [Consulta: 20/03/2025]
- [56] Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan; Kaiser, Łukasz; Polosukhin, Illia. *Attention is All You Need*. s.l.: NIPS. (Neural Information Processing series - MIT Press)/ Cornell University. 2023. [En línea]. Disponible en: <https://arxiv.org/pdf/1706.03762.pdf> [Consulta: 20/03/2025]
- [57] TensorFlow. *API Documentation*. s.l.: TensorFlow, 2024. [En línea]. Disponible en: https://www.tensorflow.org/api_docs/python/tf [Consulta: 03/01/2025]
- [58] Madrigal, Esteban. *Conoce las métricas de precisión más comunes para Modelos de Regresión*. s.l.: Grow Up, 2022. [En línea]. Disponible en: <https://www.growupcr.com/post/metricas-precision> [Consulta: 08/02/2025]

- [59] Belcic, Ivan; Stryker, Cole. *¿Qué es el ajuste de hiperparámetros?*. s.l.: IBM, 2024. [En línea] Disponible en: <https://www.ibm.com/es-es/think/topics/hyperparameter-tuning> [Consulta: 21/03/2025]
- [60] Barrero Sánchez, Sebastián Andres. *Propuesta de implementación de un sistema de BI predictivo basado en big Data y ETL avanzado para la mejora de la toma de decisiones empresariales*. s.l.: UTADEO, 2024. [En línea]. Disponible en: <https://expeditiorepositorio.utadeo.edu.co/handle/20.500.12010/36148/> [Consulta: 18/03/2025]
- [61] Çibikdiken, Ali Osman. *Comparison of ARIMA Time Series Model and LSTM Deep Learning Algorithm for Bitcoin Price Forecasting*. Prague: Research Gate/ Multidisciplinary academic conference, 2018. [En línea]. ResearchGate. Disponible en: <https://www.researchgate.net/publication/340417228> [Consulta: 23/08/2024]
- [62] McInnes, Leland; Healy, John. *Accelerated Hierarchical Density Clustering*. s.l.: Cornell University, 2017. [En línea]. Disponible en: <https://arxiv.org/pdf/1705.07321.pdf> [Consulta: 23/02/2025]
- [63] McInnes, Leland; Healy, John; Astels, Steve. *Comparing Python Clustering Algorithms*. s.l.: HDBSCAN documentación oficial, 2016. [En línea]. Disponible en: https://hdbSCAN.readthedocs.io/en/latest/comparing_clustering_algorithms.html [Consulta: 23/02/2025]
- [64] Kelleher, Jhon D.; Mac Namee, Brian; D'Arcy, Aoife. *Fundamentals of machine learning for predictive data analytics : algorithms, worked examples, and case studies*. s.l.: Cambridge, Massachusetts : The MIT Press, 2015. [En línea]. Disponible en: <https://archive.org/details/fundamentalsofma0000kell/mode/2up> [Consulta: 23/09/2024]
- [65] EITCA Academy. *¿Por qué un entrenamiento demasiado prolongado de redes neuronales conduce a un sobreajuste y cuáles son las contramedidas que se pueden tomar?*. s.l. Etica Academy, 2023. [En línea]. Disponible en: <https://es.eitca.org/artificial-intelligence/eitc-ai-dlpp-deep-learning-with-python-and-pytorch/> [Consulta: 23/04/2025]
- [66] Cui, Shunji. *A Robust Online Korean Teaching Support Technology Based on TCNN*. Jilin, China: Jilin Agricultural Science and Technology University, 2023. [En línea]. Disponible en: https://www.jmis.org/archive/view_article?pid=jmis-10-3-249 [Consulta: 23/04/2025]
- [67] Hyndman, Rob J.; Athanasopoulos, George. *Forecasting: principles and practice*. Australia: Monash University, 2021. [En línea]. Disponible en: <https://otexts.com/fpp3/nnetar.html#fig:nnet2> [Consulta: 23/04/2025]
- [68] García, Cándido. *Dashboard de predicciones del volumen de mensajes mediante RNN*. Valencia: Edicom Analytics, 2025. [En línea]. Disponible en: <https://analytics.edicomgroup.com/es/shared/RNN-Predicciones-Volumen-Transacciones/> [Consulta: 25/05/2025]
- [69] García, Cándido. *Dashboard del volumen de mensajes y su segmentación en patrones transaccionales mediante clustering*. Valencia: Edicom Analytics, 2025. [En línea]. Disponible en: <https://analytics.edicomgroup.com/es/shared/Clustering-Segmentacion-Volumen-Transacciones> [Consulta: 25/05/2025]

- [70] García, Cándido. *Dashboard del tamaño de los mensajes y su segmentación en patrones transaccionales mediante clustering*. Valencia: Edicom Analytics, 2025. [En línea]. Disponible en: <https://analytics.edicomgroup.com/es/shared/Clustering-Segmentacion-Tamaño-Transacciones> [Consulta: 25/05/2025]

CAPÍTULO 11

Anexos

11.1 Gráficas de interlocutores origen y destino de las transacciones

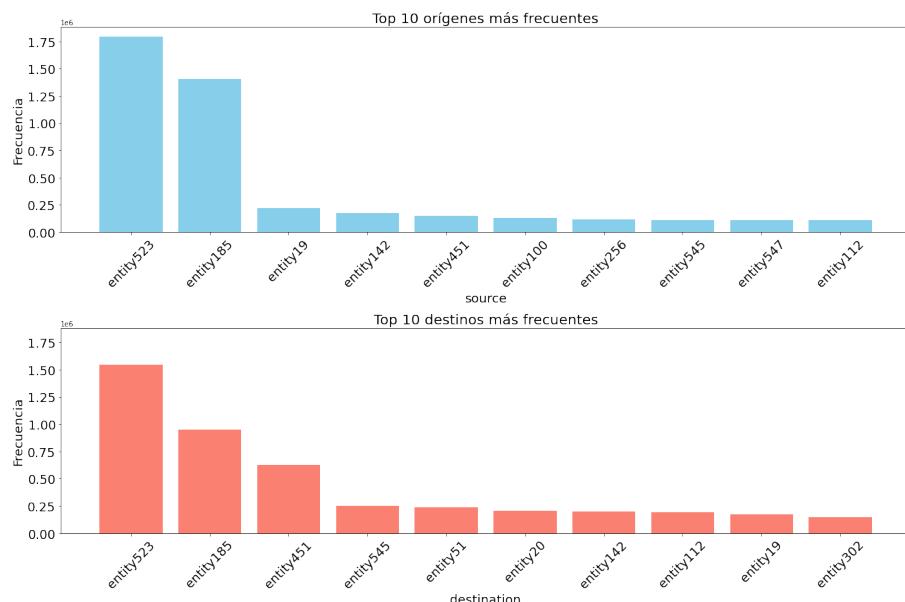


Figura 11.1: frecuencia y distribución de interlocutores de origen y destino para las transacciones según la dirección del mensaje.

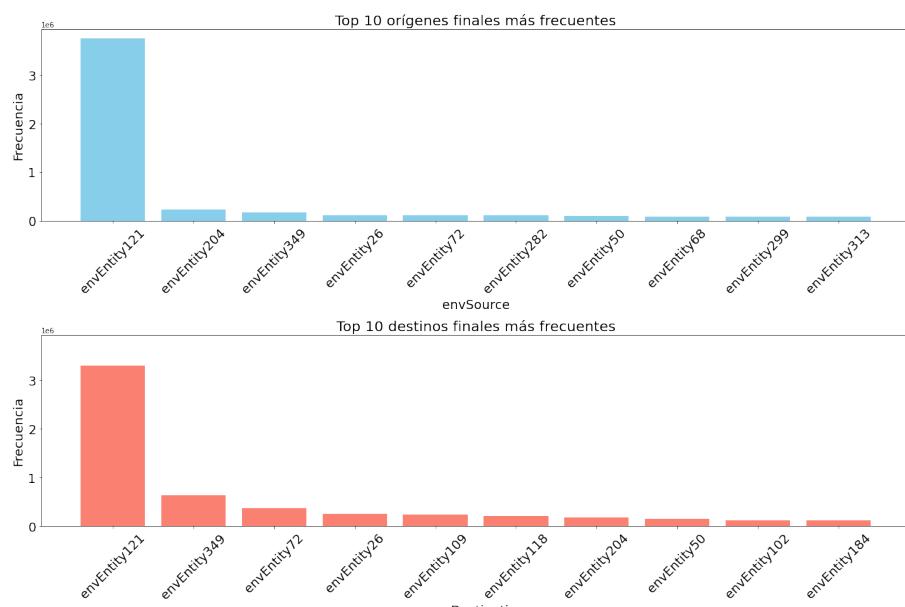


Figura 11.2: frecuencia y distribución de interlocutores de origen y destino final para las transacciones según la dirección del mensaje.

11.2 Resumen de componentes principales

A continuación se realiza una explicación más específica para las variables contribuyentes a cada una de las componentes principales adicionales que no han sido explicadas previamente.

PC4: esta componente captura la variabilidad diaria en el tamaño de los intercambios, destacando la influencia de la variable `day_of_the_week_sin`. Los tamaños de pedidos tienen una contribución positiva, mientras que las transacciones de tipo GENRALS contribuyen negativamente, generando una clara separación entre estos dos factores. Esta componente refleja la variabilidad diaria, diferenciando entre patrones regulares (tamaños de pedidos) y más irregulares (transacciones de tipo GENRALS).

PC5: esta componente representa la varianza asociada a tendencias temporales, explicada por variables como `rolling_slope` y métricas clave como el número y tamaño de los mensajes. La separación se observa en patrones horarios y diarios, con el número de mensajes de tipo OSTRPT como el principal factor determinante. Los rezagos de orden 144, 336 y 168 también juegan un papel importante. En conclusión, esta componente captura la variabilidad de las tendencias temporales, destacando el impacto de transmisiones específicas y su regularidad en el tiempo.

PC6: refleja patrones diarios en el tamaño de los mensajes y su variabilidad. Las contribuciones de `rolling_slope` y los números de mensajes asociados a ORDERS y ORSRSP son positivas, mientras que los rezagos de 177 y 48 en el número de mensajes tienen una contribución negativa. Finalmente, esta componente muestra cómo las variaciones diarias en el tamaño y la pendiente de las transacciones determinan su variabilidad, especialmente entre entidades clave.

PC7: esta componente se enfoca en ratios acumulativos (por tipo de mensaje, tamaño y categoría) y promedios móviles considerando varios rezagos. Captura la consistencia acumulativa de diversas métricas, destacando transmisiones de tipo ORDERS y tamaños de mensajes medios a altos. No captura patrones temporales claros. En definitiva, esta componente representa la estabilidad en métricas homogéneas a lo largo del tiempo, sin capturar patrones temporales específicos.

PC8: contribuciones exclusivamente positivas, centradas en las diferencias entre períodos consecutivos, especialmente mediante rezagos y pendientes móviles. Sobresalen los mensajes de tipo OSTRPT, transacciones sin protocolo de seguridad criptográfica (NO_CRYPTO) y mensajes de tamaños bajos o medio-bajos. En síntesis, esta componente destaca las variaciones a corto plazo, con separación clara basada en transacciones de menor tamaño y tipos de mensajes específicos.

PC9: relacionado con cambios diarios, mediante los rezagos de orden 24 en el tamaño de las transacciones y ratios acumulativos. También presenta patrones de intercambios ORDERS y contribuciones negativas relacionadas con `day_of_the_week_sin`. Esta componente representa un contraste entre estabilidad y variabilidad en los cambios diarios del tamaño de los mensajes, influenciado por la tasa de envío y el número de mensajes relacionados con pedidos.

PC10: esta componente refleja patrones cíclicos semanales, resaltando la periodicidad en el tamaño de los mensajes (Medium Low, Medium High) y las interacciones regulares entre interlocutores clave. Los rezagos de orden 148, 316, 484 y 652 también tienen un impacto relevante. En conclusión, esta componente representa las tendencias regulares en la comunicación semanal, con un enfoque en las interacciones consistentes y tamaños de mensaje intermedios como patrones diferenciadores.

11.3 Integración de los datos en las herramientas de Edicom

En este apartado se describe cómo se han integrado los resultados obtenidos a partir de los modelos desarrollados en las herramientas de la empresa Edicom. En primer lugar se detalla el flujo de integración con la herramienta de **LTA**, que permite a **Edicom Analytics** acceder a dichos datos y generar a partir de ellos visualizaciones que pueden ser utilizadas para la presentación de los resultados de una forma clara y dirigida.

Para la carga, es requerida la conversión de nuestros datos, que inicialmente están en formato .csv al formato estándar aceptado por **LTA**. Esta carga y conversión se realizan a través de **IPaaS** utilizando procesos de conversión desarrollados en **EbiMap**. Exceptuando el primer paso de carga de los datos, el flujo se ha automatizado mediante un sistema de publicaciones y suscripciones activadas por reglas, mediante las cuales scripts de lanzamiento de procesos de transformación de formato son habilitados.

En primer lugar, se importa en **IPaaS** el conjunto de datos resultante del modelado, utilizando el esquema CSV_KIBANA_LOGS, activando la ejecución del script ENVIAR EBI-EBI-JS CSV_TO_EDICOMSTATS descrito en el listing 11.1, donde se aplica la transformación LANZADOR_DATOS_CSV mediante la función EnviarEBIEBI. Una vez completada la transformación, se publicará el resultado con el esquema XML_EDICOMSTATS.

La transformación, realizada mediante el proceso de transformación de EbiMap LANZADOR_DATOS_CSV, convierte los datos importados en formato CSV al formato estándar de métricas definido por Edicom.

```
1. Incluir funciones o módulos externos necesarios para ejecutar transformaciones y
   publicar los resultados en la herramienta de Ipaas :
   IncluirModulo("ENVIAR EBI--EBI--JS")
2. Definir la aplicación destino del proceso en la herramienta de Ipaas:
   AplicacionDestino = "CAGARCIAT"
3. Ejecutar el proceso de transformación de datos para:
   - Convertir los datos CSV en un formato estándar XML usando el mapa
     "LANZADOR_DATOS_CSV".
   - Publicar el resultado con el esquema "XML_EDICOMSTATS".
     TransformarYEnviarDatos(EsquemaSalida = "XML_EDICOMSTATS",
                               OrigenDatos = "CSV_KIBANA_LOGS",
                               MapaTransformacion = "/EDICOMSTATS_TFG/LANZADOR_DATOS_CSV"
                           )
4. Fin del proceso.
```

Listing 11.1: script ENVIAR EBI-EBI-JS CSV_TO_EDICOMSTATS que transforma los documentos CSV al formato estandarizado EDICOMSTATS y pública de nuevo en IPaaS bajo el esquema XML_EDICOMSTATS.

Debido al gran volumen de datos, el archivo .csv resultante se divide en subarchivos de 1000 GB para evitar bloqueos durante el procesamiento. Estos posteriormente se procesan mediante un mapa transformador.

En el formato Edicomstats, cada nodo <Metric> traslada información relevante como identificadores, el nombre del documento a cargar, fecha, y fecha de archivo hasta. Dentro de cada segmento Metric existen nodos Data que contienen secciones de metadatos, donde cada subelemento Entry representa un metadato específico del documento. Por otro lado, el nodo Etiquetas incluye las claves para la asociación en **LTA** y **Edicom Analytics**. Cada etiqueta tiene dos atributos: *key*, que define el nombre del metadato, y *entryType*, que especifica el tipo de dato asociado.

Los tipos de datos que pueden existir son:

- **NA:** no analizable.

- DT: fecha.
 - S: cadena de texto.
 - I: número entero.
 - D: número decimal.
 - L: número largo.

Una vez que los datos se han convertido al formato estándar y se ha publicado el esquema XML_EDICOMSTATS (figura 11.3), se lanza un script TEMPLATE_SEND_IPAAS_TO_LTA, descrito en el listing 11.2 para realizar la transformación al formato XML_DOCUMENTOS (figura 11.4), que es el aceptado en LTA. Utilizando las credenciales necesarias, se solicita un token de acceso a la plataforma con el que se conecta al servicio y se insertan los metadatos mediante la función ReceiveFromIPAASAndPostLTA.

Figura 11.3: ejemplo de fichero con formato xml_EdicomStats.

Este proceso es estándar y todas las funciones relacionadas con la transformación de XML_EDICOMSTATS a XML_DOCUMENTOS estaban previamente definidas.

Listing 11.2: script TEMPLATE_SEND_IPAAS_TO_LTA que solicita un token de autenticación y, si es válido, transforma y envía datos desde iPaaS al sistema LTA en formato XML DOCUMENTS.

Mediante un flujo de publicaciones y suscripciones que representan los mensajes y causan el lanzamiento automático los procesos en la plataforma, los metadatos se cargan en la herramienta **LTA**, como se muestra en la figura 11.5.

```
<?xml version='1.0' encoding='UTF-8'?>
<document><Document xmlns='http://www.w3.org/2001/XMLSchema' xmlns:document='http://www.edicongroup.com/schemas/elite/1.0/documents'>
<document:Document id='67268931' name='Thic176e-897e-11ed-9572-5db8a10e7c-A28017895_PRV_classification' compression='NONE' content-type='text/plain' />
<document:Metadata action='Create' />
<document:Entry key='Day_of_week_NA' value='1' type='String'/>
<document:Entry key='DATEINSERT_DT' value='2023-01-01T03:59:00.000Z' type='DateTime'/>
<document:Entry key='ENVIRONMENT_NA' value='A28017895_PRV' type='String'/>
<document:Entry key='ENVDESTINATION_NA' value='8422416000016' type='String'/>
<document:Entry key='FILE_NAME' value='9ccf0a95-89ba-11ed-989b-f70276e3c4ee-2301010957047222097-A28017895_PRV' type='String'/>
<document:Entry key='IEPE_NA' value='EDIONLINE' type='String'/>
<document:Entry key='MESSAGE_ID' value='67268931' type='String'/>
<document:Entry key='SIZE_D' value='12304.000000000002' type='Double'/>
<document:Entry key='SIZE_Z_SCORE_D' value='2945151543398919' type='Double'/>
<document:Entry key='SIZE_CAT_NA' value='Medium_High' type='String'/>
<document:Entry key='SIZE_DEV_FROM_MEAN_D' value='-46.29015060730165' type='Double'/>
<document:Entry key='SIZE_OF_WEEK_NA' value='6' type='String'/>
</document><document id='67268941' name='0ef5f64d-897e-11ed-9572-5db8a10e7c-A28017895_PRV' value='0ef5f64d-897e-11ed-9572-5db8a10e7c-A28017895_PRV_classification' compression='NONE' content-type='text/plain' />
<document:Document id='67268941' name='application/vnd.edicom.EDICOM_TRAINING.statsfgclassification' compression='NONE' content-type='text/plain' />
```

Figura 11.4: ejemplo del fichero en formato xml_documentos.

La captura de pantalla muestra la interfaz de usuario de la herramienta LTA. En la parte izquierda, hay un menú lateral con opciones como 'Documentos', 'Recientes', 'Filtros', 'Tipos' y 'Metadatos'. La lista principal muestra 26 coincidencias. Cada elemento de la lista incluye un icono, el ID del documento, su nombre y su tipo de archivo. Una flecha roja apunta a la lista de resultados. A la derecha, se abre una vista previa detallada de un documento, mostrando una tabla de metadatos. La tabla tiene columnas para 'Nombre', 'Valor' y otras columnas que no están completamente visibles. Los datos incluyen información como la fecha de inserción, el destino, el tamaño y el tipo de archivo.

Figura 11.5: visualización de los documentos y sus metadatos en LTA.

11.4 Edicom Analytics

Una vez integrados los datos mediante el proceso descrito en el apartado anterior, la información queda disponible para su consulta desde la herramienta **Edicom Analytics**.

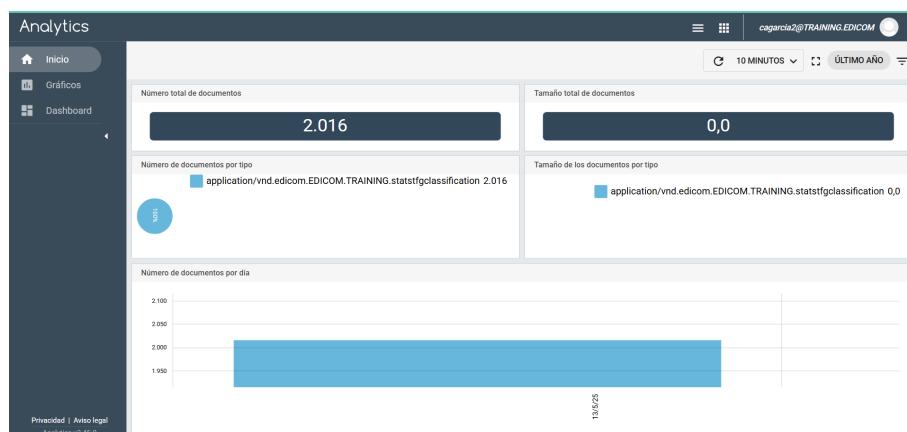


Figura 11.6: visión global de Edicom Analytics.

Al acceder a la aplicación, se presentan tres secciones principales. La primera es la sección de Inicio, mostrada en la figura 11.6, donde se visualiza un panel general que ofrece información agregada sobre los documentos disponibles para el usuario.

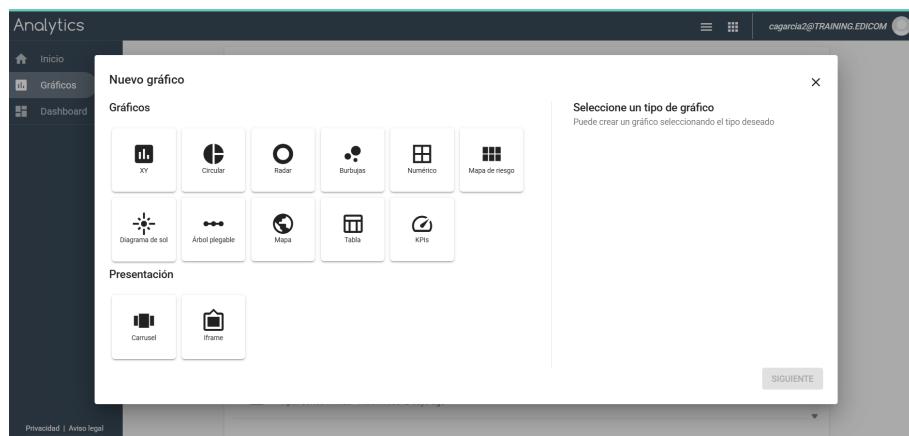


Figura 11.7: creación de una gráfica partiendo de cero y sus posibles tipos.

En la sección de gráficas, es posible crear una visualización desde cero. Para ello, el sistema solicita primero definir el tipo de gráfica y asignarle un nombre, antes de acceder a la pantalla de configuración. Tal como se muestra en la figura 11.7, las opciones disponibles incluyen: gráfico XY, gráficos circulares, de radar, de burbujas, numéricos, mapas de riesgo, diagramas de sol, árboles plegables, mapas, tablas de datos y KPIs. Estas visualizaciones pueden presentarse en formato carrusel o en modo frame, facilitando una exposición dinámica y estructurada de la información.

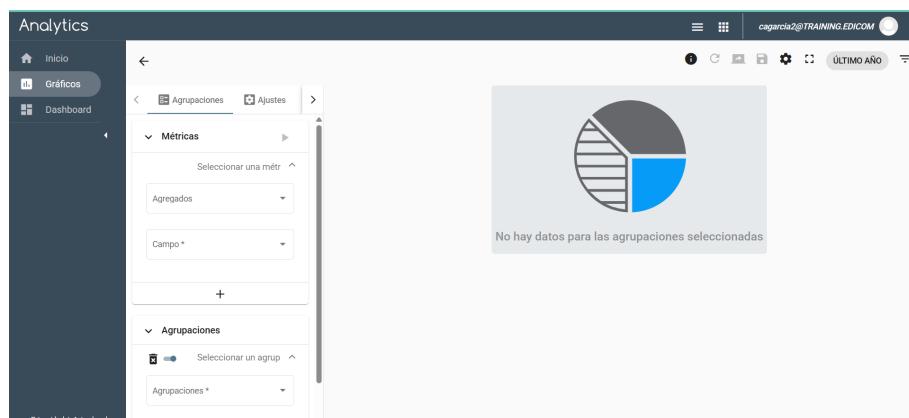


Figura 11.8: posibles ajustes durante la creación de una gráfica.

Una vez en la pantalla de configuración (figura 11.8), se presentan distintas secciones para ajustar el resultado del gráfico. Estas opciones son las siguientes:

- **Agrupaciones:** esta es la sección principal donde se definen la fuente de datos, las métricas a evaluar y las funciones de agrupación, para la cual existen las siguientes opciones disponibles: suma, conteo, media, máximo, mínimo y número de valores únicos.

Asimismo, se puede configurar la función de agrupación, que permite especificar los criterios bajo los cuales se agruparán los datos. Entre las opciones se incluyen, el histograma y el histograma de fechas. Finalmente, el número y tipo de agrupaciones disponibles dependerá del tipo de gráfica seleccionada durante la configuración.

- **Ajustes:** configuración visual del gráfico. (e.g Título, leyenda, ejes...etc)
- **Traducciones:** opciones de traducción de las etiquetas de la gráfica en distintos idiomas.

- **Información:** título, descripción, etiquetas y permisos.

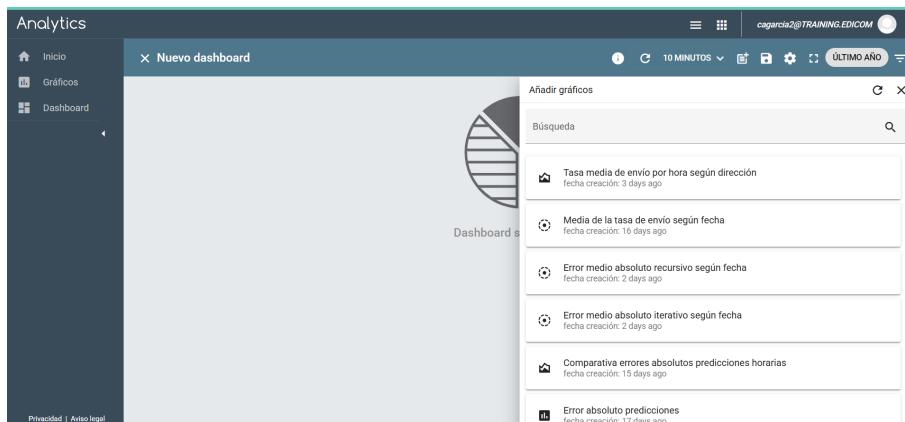


Figura 11.9: creación de un dashboard desde cero.

Pasamos a la tercera y última sección, dedicada a la creación de dashboards. Una vez generadas las distintas gráficas, es posible agruparlas para presentar la información de forma integrada. En esta sección, el usuario puede consultar, editar dashboards existentes o crear nuevos desde cero.

Al crear un dashboard, se solicita inicialmente seleccionar el tipo de visualización, ya sea una secuencia de gráficas o un dashboard simple. Tras asignarle un nombre, el usuario puede añadir las gráficas ya creadas que desee, colocándolas libremente en el tablero y ajustando sus dimensiones según sea necesario (figura 11.11). En general, se trata de una herramienta intuitiva y fácil de utilizar, diseñada para mostrar visualizaciones de forma clara y eficiente.

11.4.1. Generación de gráficas de ejemplo con Edicom Analytics

Con el objetivo de facilitar la interpretación y el aprovechamiento de los resultados generados por los modelos desarrollados, se han creado tres dashboards interactivos. Estas visualizaciones permiten explorar de forma intuitiva y detallada las principales conclusiones del proyecto. A continuación, se presentan cada uno de ellos:

- **Dashboard de las predicciones del volumen de transacciones:**

Este dashboard, del cual una parte puede ser visualizado en la figura 11.10, muestra las predicciones del volumen de mensajes transaccionados en la plataforma, realizadas mediante modelos de redes neuronales recurrentes (GRU). Se presentan gráficos que comparan las predicciones tanto con el método iterativo como con el método recursivo, frente a los valores reales observados. Además, se incluye el cálculo del error medio absoluto para evaluar la precisión del modelo. Las visualizaciones están segmentadas por variables temporales como día de la semana, mes del año, y si se trata de un fin de semana, así como por horas del día, permitiendo analizar patrones y variaciones en la tasa y número de mensajes en función del tipo y categoría del mensaje, y la dirección de la transacción.



Figura 11.10: dashboard del volumen de mensajes predicho mediante el modelo de redes neuronales. [68]

- **Dashboard de los clusters según el volumen de las transacciones:**

Este dashboard, del cual una parte puede ser visualizado en la figura 11.11, presenta los resultados del análisis de clustering realizado mediante K-Means, centrado en la segmentación por horas de intercambios según el número de mensajes procesados. Las visualizaciones permiten filtrar por fecha, hora, día de la semana y tipo de día (laborable o fin de semana), mostrando cómo se agrupan los diferentes patrones de uso. Esto facilita la identificación de perfiles de comportamiento y tendencias específicas dentro del volumen de transmisiones.



Figura 11.11: dashboard del volumen de mensajes y su segmentación en patrones transaccionales mediante técnicas de clustering. [69].

- **Dashboard de los Clusters según el tamaño de las transacciones:**

Complementario al anterior, este dashboard del cual una parte puede ser visualizado en la figura 11.12, se enfoca en la segmentación de intercambios basada en el tamaño de los mensajes. Se presentan visualizaciones similares que permiten observar la distribución y clasificación de los clusters en función del volumen y características de las transacciones, también filtrables por variables temporales.

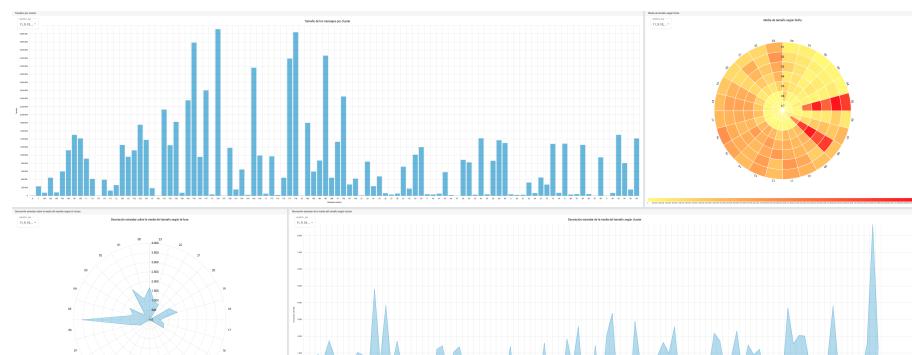


Figura 11.12: dashboard del tamaño de los mensajes y su segmentación en patrones transaccionales mediante técnicas de clustering. [70].

Los tres dashboards incorporan una variedad de visualizaciones, como series temporales, gráficos de barras, de área y gráficos radar, con el objetivo de proporcionar una experiencia visual clara, atractiva y accesible para el usuario. Además, incluyen funcionalidades interactivas emergentes de información (*tooltips*) con información detallada, la posibilidad de activar o desactivar métricas específicas, y filtros avanzados que permiten analizar los datos desde distintas perspectivas.

Estas visualizaciones representan una primera aproximación al potencial de los modelos desarrollados, y ponen en evidencia su aplicabilidad práctica. Gracias a su diseño flexible, los dashboards y los datos, pueden ampliarse y adaptarse fácilmente a futuras necesidades, incorporando nuevas métricas o dimensiones que enriquezcan el análisis y faciliten una interpretación aún más precisa de los resultados obtenidos.

11.5 Objetivos de desarrollo sostenible

A continuación, el grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				
ODS 5. Igualdad de género.			X	
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.			X	
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.		X		

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

El presente Trabajo Fin de Grado aborda el desarrollo y la aplicación de técnicas avanzadas de análisis de datos en un entorno empresarial real, con un enfoque orientado a la segmentación y detección de patrones para mejorar procesos internos y optimizar la toma de decisiones. Mediante la implementación de modelos predictivos y algoritmos de clustering sobre datos de transacciones EDI, se espera generar un impacto tangible tanto en la eficiencia operativa como en la gestión del conocimiento dentro de la empresa Edicom. Esta iniciativa representa un claro ejemplo de cómo la innovación tecnológica puede modernizar infraestructuras digitales y fortalecer la competitividad en sectores estratégicos. Al mismo tiempo, contribuye a la mejora de las condiciones laborales y al crecimiento sostenible, promoviendo procesos más ágiles y efectivos. La gestión eficiente de los recursos digitales, basada en analítica avanzada, permite reducir desperdicios informacionales y mejorar la calidad de los servicios ofrecidos, lo cual implica una utilización más responsable y racional de los sistemas disponibles. Por otro lado, el desarrollo del proyecto se ha apoyado en una estrecha colaboración entre distintos perfiles profesionales desde desarrolladores hasta analistas y usuarios finales, lo que pone de manifiesto el valor del trabajo conjunto y la coordinación entre áreas para alcanzar resultados de alto impacto. En suma, este trabajo no solo demuestra el potencial de la inteligencia artificial y el aprendizaje automático aplicados a entornos SaaS, sino que evidencia cómo la tecnología, cuando se orienta de manera estratégica, puede ser una herramienta clave para avanzar hacia un desarrollo más eficiente, sostenible y colaborativo.