UNIVERSIDAD DE ANTIOQUIA

1 8 0 3

Análisis Predictivo y Detección de Anomalías en el Consumo de MIPS del Grupo Éxito: Optimización y Eficiencia Operacional

Estiben Andrey González Londoño

Universidad de Antioquia
Facultad de Ciencias Exactas y Naturales
Instituto de Matemáticas
Medellín, Colombia
2025

Análisis Predictivo y Detección de Anomalías en el Consumo de MIPS del Grupo Éxito: Optimización y Eficiencia Operacional

Estiben Andrey González Londoño

Trabajo de grado presentado como requisito parcial para optar al título de:
ESTADÍSTICO

Santiago Echeverri Valencia
Orientador, Instituto de Matemáticas

John Wilmerckx Quintero Pinzon
Orientador Externo, Grupo Éxito

Universidad de Antioquia
Facultad de Ciencias Exactas y Naturales
Instituto de Matemáticas
Medellín, Colombia
2025

A mi madre, cuyo sacrificio y apoyo incondicional hicieron posible el desarrollo completo de mi carrera, y cuyo ejemplo me ha nutrido como persona. A mi padrastro, quien, sin obligación alguna, respaldó mi camino académico y personal, permitiéndome avanzar con seguridad y determinación. Sin ellos, no sería la mitad de la persona que soy hoy.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a las personas e instituciones que hicieron posible la realización de este trabajo de grado.

A mi asesor interno, **Santiago Echeverri Valencia**, por su invaluable aporte a mi formación académica. Sus cursos, enfocados en la aplicación práctica de las matemáticas y la estadística en el entorno laboral, fueron fundamentales para el desarrollo de este trabajo. Su enfoque aplicado me permitió afrontar con mayor claridad los desafíos analíticos y metodológicos que surgieron a lo largo del proyecto.

A mi asesor externo, **John Wilmerckx Quintero Pinzón**, por su orientación y guía en este primer acercamiento al entorno corporativo. Su experiencia y consejos fueron esenciales para comprender la estructura de los proyectos en el ámbito empresarial, la importancia de la colaboración entre equipos y las metodologías adecuadas para la ejecución de avances dentro de una organización.

A la **Universidad de Antioquia**, por ser la institución que me brindó las herramientas y el conocimiento necesarios para alcanzar este logro.

A **Grupo Éxito**, por abrirme las puertas y brindarme la oportunidad de aplicar mis conocimientos en un entorno real, permitiéndome crecer profesionalmente y desarrollar este proyecto con un impacto significativo.

Resumen

El consumo de Millones de Instrucciones Por Segundo (MIPS) es un factor crítico en la eficiencia operativa de Grupo Éxito, dado su impacto en la infraestructura tecnológica y los costos asociados. En este trabajo, se relata el desarrollo de un microservicio diseñado para optimizar el análisis del consumo de MIPS mediante la detección de anomalías y la predicción del uso futuro de los recursos. Se implementaron modelos avanzados de análisis de datos en *Python*, incluyendo técnicas de detección de valores atípicos basadas en el rango intercuartílico y la desviación absoluta mediana, así como modelos de predicción utilizando la librería `PROPHET`. Para garantizar una integración eficiente con los sistemas de la compañía, se construyó una arquitectura modular que permite la extracción, procesamiento y almacenamiento de datos de una base de datos de origen a una base de datos normalizada. La información procesada se visualiza en un tablero interactivo de *Power BI*, proporcionando una herramienta estratégica para el monitoreo y la toma de decisiones. Adicionalmente, se implementó un flujo de despliegue automatizado mediante *Azure DevOps*, *Kubernetes* y *Docker*, asegurando la estabilidad y escalabilidad del sistema en entornos productivos. Los resultados obtenidos no solo optimizan el uso de MIPS, sino que también sientan las bases para futuras mejoras en la gestión de infraestructura tecnológica del Grupo Éxito. Este trabajo demuestra el potencial del análisis predictivo y la automatización en la gestión eficiente de recursos computacionales, proponiendo un modelo replicable para otras organizaciones con necesidades similares.

Palabras clave: MIPS, detección de anomalías, predicción de consumo, PROPHET, IQR, MAD, normalización de datos, Power BI, Azure DevOps, Kubernetes, Docker, automatización, gestión de infraestructura, optimización de recursos.

Abstract

The consumption of Millions of Instructions Per Second (MIPS) is a critical factor in the operational efficiency of Grupo Éxito, given its impact on the technological infrastructure and the associated costs. In this paper, we report the development of a microservice designed to optimize the analysis of MIPS consumption by detecting anomalies and predicting future resource usage. Advanced data analysis models were implemented in *Python*, including outlier detection techniques based on interquartile range and median absolute deviation, as well as prediction models using the `PROPHET` library. To ensure efficient integration with the company's systems, a modular architecture was built to allow data extraction, processing, and storage from a source database to a normalized database. The processed information is visualized in an interactive Power BI dashboard, providing a strategic tool for monitoring and decision-making. Additionally, an automated deployment flow was implemented using Azure DevOps, Kubernetes, and Docker, ensuring the stability and scalability of the system in productive environments. The results obtained not only optimize the use of MIPS, but also lay the foundations for future improvements in the management of the technological infrastructure of Grupo Éxito. This work demonstrates the potential of predictive analytics and automation in the efficient management of computational resources, proposing a replicable model for other organizations with similar needs.

Keywords: MIPS, anomaly detection, consumption prediction, PROPHET, IQR, MAD, data normalization, Power BI, Azure DevOps, Kubernetes, Docker, automation, infrastructure management, resource optimization.

Contenido

1. Introducción

El Grupo Éxito, líder del sector retail en Colombia, opera con sistemas tecnológicos robustos que garantizan el funcionamiento de su infraestructura operativa diaria. Entre estos, destaca SINCO (Sistema de Información Comercial), el núcleo tecnológico que soporta la gestión comercial de la compañía. Este sistema se ejecuta sobre la plataforma *ClearPath*, diseñada para grandes organizaciones que requieren alta disponibilidad y rendimiento en el manejo de volúmenes masivos de datos y transacciones [20]. El consumo de recursos en esta plataforma se mide en MIPS (Millones de Instrucciones Por Segundo), una métrica crítica para la empresa, pues cada unidad representa un costo de 95 dólares. En febrero de 2021, Grupo Éxito firmó un contrato con Unisys—proveedor del servicio *ClearPath*—para el uso de 281,400 MIPS, con un valor cercano a los 27 millones de dólares.

Dado el impacto financiero de este recurso, la optimización del consumo de MIPS se convirtió en una prioridad estratégica. Sin embargo, el Equipo de Gestión de MIPS enfrentaba limitaciones significativas en sus herramientas de monitoreo y análisis. La primera barrera era la detección rápida y precisa de los procesos dentro de SINCO que, debido a un mayor número de ejecuciones o tiempos de procesamiento prolongados, generaban picos de consumo inesperados. Con cerca de diez mil procesos activos, identificar aquellos responsables de incrementos anómalos era una tarea compleja que exigía rapidez y precisión para alertar a los equipos encargados de mitigar su impacto. La segunda necesidad clave radicaba en la proyección del consumo futuro: el equipo buscaba entender cómo los patrones históricos y la tendencia actual podían utilizarse para prever el agotamiento de los MIPS disponibles y anticipar posibles medidas correctivas. Estas problemáticas dieron origen a la necesidad de desarrollar nuevas herramientas analíticas que permitieran un monitoreo más eficiente y una planificación estratégica fundamentada en datos.

Las dos limitaciones previamente expuestas constituyen los ejes centrales de este trabajo. Para abordarlas, se estableció como objetivo el desarrollo de un microservicio siguiendo el marco Scrum de gestión de proyectos. En la fase de modelado, se decidió utilizar Python para implementar métodos de detección de valores atípicos en el consumo de MIPS, empleando el Rango Intercuartílico y la Desviación Absoluta Mediana. Para la predicción del consumo futuro, se seleccionó la metodología PROPHET, diseñada para modelar series de tiempo capturando tendencias, patrones y estacionalidades. El modelo debía generar estimaciones con sus respectivos intervalos de confianza y garantizar que las métricas de error—RMSE, MAE, MAPE y sMAPE—se mantuvieran por debajo del 15 %.

El despliegue del microservicio debía alinearse con las pautas establecidas por el área de desarrollo de *software* de la compañía, utilizando Azure DevOps. Además, se estableció que la ejecución del microservicio fuera automática y diaria desde los servidores productivos de Grupo Éxito. Finalmente, toda la información generada debía integrarse en los tableros de Power BI habilitados por los sistemas productivos de la empresa, asegurando una visualización accesible y efectiva para todos los actores involucrados en la toma de decisiones.

Esta sección inicia con una contextualización dentro de las áreas de la ciencia, la tecnología y la ingeniería involucradas, y concluye con la justificación de la selección metodológica. En segundo lugar, se detalla la metodología de trabajo, organizada en una secuencia lineal con posibilidad de iteración sobre el cronograma de actividades. Posteriormente, se exponen los resultados obtenidos en cada fase, acompañados de un análisis crítico y una discusión de los hallazgos. Finalmente, el documento cierra con las conclusiones, donde se resumen los logros alcanzados, se reconocen las limitaciones del estudio y se proponen posibles líneas de investigación futura.

Más allá de ofrecer soluciones analíticas para optimizar el consumo de MIPS, este trabajo buscó contribuir a la continuidad y eficiencia operativa del Grupo Éxito, en concordancia con su estrategia de gestión tecnológica sostenible hasta, al menos, el segundo semestre de 2028. Se espera que los resulta-

dos obtenidos no solo mejoren el control y la previsión del uso de recursos, sino que también sienten las bases para futuras innovaciones en la administración de infraestructura tecnológica, fortaleciendo así la sostenibilidad financiera y operativa de la compañía.

2. Marco Teórico

2.1. Marco Scrum para Gestión de Proyectos

Scrum es un marco ligero que permite a equipos y organizaciones generar valor mediante soluciones adaptativas para problemas complejos. Se fundamenta en el empirismo y el pensamiento Lean. El empirismo sostiene que el conocimiento surge de la experiencia y la toma de decisiones debe basarse en la observación, mientras que el pensamiento Lean busca reducir desperdicios y enfocarse en lo esencial. Este marco utiliza un enfoque iterativo e incremental para mejorar la previsibilidad y gestionar riesgos [15].

2.1.1. Conceptos Clave

- Objetivo del Producto: Meta a largo plazo del equipo Scrum que guía la planificación del trabajo. Define el propósito del producto y debe cumplirse antes de asumir un nuevo objetivo.
- Trabajo Pendiente del Producto: Lista estructurada y cuidada meticulosamente que utiliza el Propietario del Producto para guiar las tareas del equipo de desarrollo [13]. Es la única fuente de trabajo del equipo Scrum. Se refina continuamente para descomponer los elementos en tareas más pequeñas y detalladas. Los encargados del proyecto deben reordenar y actualizar con frecuencia el Trabajo Pendiente del Producto, en función de la nueva información y la lista de requisitos que obtengan de los clientes, del mercado o del equipo del proyecto [11].
- *Sprint:* un *Sprint* es un período de tiempo fijo, generalmente de una a cuatro semanas, durante el cual el equipo Scrum trabaja para completar un Incremento del producto utilizable y de alta calidad.
- Objetivo del *Sprint*: Propósito único del *Sprint* que alinea al equipo en una dirección común. Proporciona flexibilidad en la ejecución y se ajusta según el avance del trabajo sin afectar su propósito general.
- Trabajo Pendiente del *Sprint*: Plan de trabajo para el *Sprint*, compuesto por el Objetivo del *Sprint*, los elementos del Trabajo Pendiente del Producto seleccionados y un plan detallado para entregar un incremento funcional. Se actualiza continuamente según el progreso del equipo.
- Incremento: Entregable funcional y verificable que contribuye al Objetivo de Producto. Cada incremento se suma a los anteriores y puede ser entregado antes del fin del *Sprint* si está terminado y es utilizable.

2.1.2. Actores

- Maestro Scrum: Líder y facilitador del equipo Scrum. Garantiza la correcta aplicación del marco de trabajo Scrum, fomenta la mejora continua y elimina impedimentos. Actúa como mentor para el equipo, el propietario del producto y la organización en la adopción de Scrum.
- Propietario del Producto: Responsable de maximizar el valor del producto y gestionar eficazmente el trabajo pendiente del producto. Define y comunica los objetivos del producto, prioriza las tareas y garantiza la transparencia en el trabajo pendiente.

- Desarrolladores: Miembros del equipo responsables de construir un incremento funcional en cada *Sprint*. Se encargan de planificar el *Sprint*, garantizar la calidad del producto, adaptar su trabajo diariamente y colaborar profesionalmente.

Los *Sprints* son el pilar central del marco Scrum, donde las ideas se transforman en valor. Tienen una duración fija de hasta un mes, garantizando consistencia en el desarrollo. Un nuevo *Sprint* comienza inmediatamente después de finalizar el anterior. Dentro de cada *Sprint* se integran cuatro eventos clave:

- La planificación del *Sprint*
- Scrum diario
- Revisión del *Sprint*
- La retrospectiva del *Sprint*

2.1.3. Planificación del *Sprint*

El *Sprint* inicia estableciendo el trabajo que se realizará. El plan resultante es creado de manera colaborativa por todo el equipo de Scrum. De esta planificación surge el Trabajo Pendiente del *Sprint*.

El Propietario del Producto se asegura de que los asistentes estén preparados para discutir los elementos más importantes del Trabajo Pendiente del Producto y cómo estos se alinean con el Objetivo del Producto. Además, el equipo de Scrum puede invitar a otras personas para proporcionar asesoramiento.

La planificación del *Sprint* aborda los siguientes temas:

1. **¿Por qué este *Sprint* es valioso?**

   El Propietario del Producto propone cómo el producto podría aumentar su valor y utilidad en el *Sprint* actual. Luego, el Equipo Scrum colabora para definir un Objetivo de *Sprint*, que comunica el valor del *Sprint* a las partes interesadas. Este objetivo debe definirse antes de finalizar la planificación.

2. **¿Qué se puede hacer en este *Sprint*?**

   A través del debate con el Propietario del Producto, los desarrolladores seleccionan los elementos del Trabajo Pendiente del Producto que formarán parte del *Sprint* actual. Durante este proceso, pueden refinar estos elementos para mejorar la comprensión y confianza.

3. **¿Cómo se realizará el trabajo elegido?**

   Para cada elemento seleccionado del Trabajo Pendiente del Producto, los desarrolladores planifican el trabajo necesario para generar un Incremento. Esto generalmente implica descomponer los elementos en tareas más pequeñas que puedan completarse en un día o menos. La decisión de cómo realizar este trabajo recae completamente en los desarrolladores; nadie más les indica cómo convertir los elementos del Trabajo Pendiente del Producto en Incrementos de valor.

El Objetivo del *Sprint*, los elementos del Trabajo Pendiente del Producto seleccionados para el *Sprint* y el plan para entregarlos se conocen colectivamente como el trabajo pendiente del *Sprint*.

2.1.4. Scrum diario

El Scrum diario es una reunión de 15 minutos donde los desarrolladores inspeccionan el progreso hacia el Objetivo del *Sprint* y ajustan el Trabajo Pendiente del *Sprint* según sea necesario. Si el Propietario del Producto o el Maestro Scrum trabajan en elementos del Trabajo Pendiente del *Sprint*, pueden participar como desarrolladores.

Los desarrolladores pueden elegir cualquier estructura o técnica, siempre que el Scrum diario se enfoque en el progreso y genere un plan accionable para el siguiente día de trabajo, promoviendo la autogestión y el enfoque.

2.1.5. Revisión del *Sprint*

La revisión del *Sprint* tiene como propósito inspeccionar los resultados obtenidos durante el *Sprint* y definir posibles adaptaciones futuras. Durante este evento, el Equipo Scrum presenta el trabajo realizado a las partes interesadas y se evalúa el progreso hacia el objetivo del Producto.

En esta sesión, el equipo de Scrum analiza los logros alcanzados en el *Sprint* y los cambios en el entorno que puedan influir en el desarrollo del producto. Con base en esta información, los asistentes colaboran para determinar los próximos pasos, y el Trabajo Pendiente del Producto puede ajustarse en función de nuevas oportunidades o prioridades emergentes.

2.1.6. Retrospectiva del *Sprint*

La retrospectiva del *Sprint* marca la conclusión del *Sprint* y tiene como propósito identificar oportunidades de mejora para aumentar la calidad y eficacia del equipo. Durante este evento, el Equipo Scrum inspecciona el desarrollo del último *Sprint*, analizando aspectos clave como las interacciones entre los miembros, los procesos utilizados y las herramientas empleadas.

Los elementos evaluados pueden variar según el contexto del trabajo. Se identifican las suposiciones que llevaron a desviaciones, explorando sus causas y repercusiones. Asimismo, el equipo reflexiona sobre los aspectos que funcionaron bien, los problemas que surgieron y la manera en que estos fueron —o no fueron— resueltos.

Como resultado de esta sesión, se establecen las mejoras más relevantes para optimizar el desempeño del equipo. Aquellas con mayor impacto se implementan de inmediato y, si es necesario, pueden incorporarse al Trabajo Pendiente del *Sprint* del próximo *Sprint*.

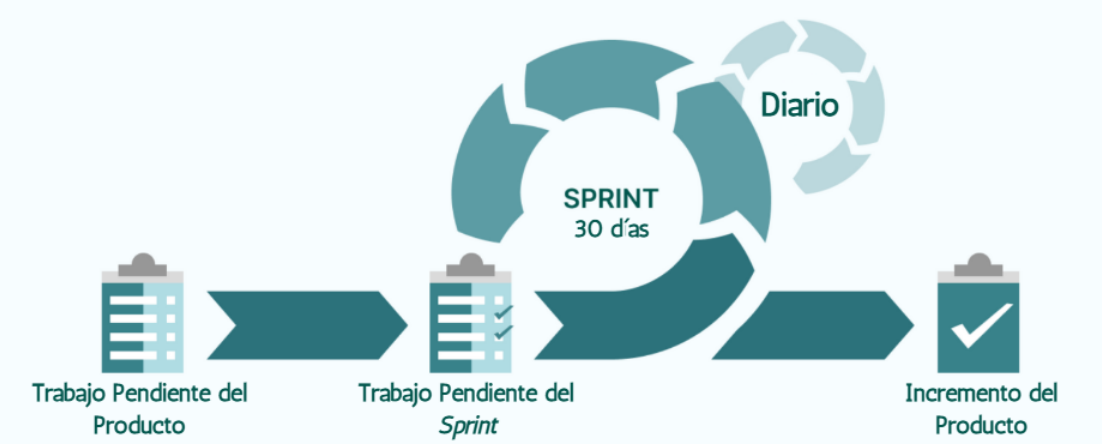

Figura 1: Flujo de Trabajo del Marco Scrum.

2.2. Análisis Exploratorio de Datos

El análisis exploratorio de datos (AED) es el primer paso en el análisis de datos, ya que explora las características básicas observadas en un conjunto de datos y proporciona directrices y diagnósticos para el modelado estadístico [12]. Desde el trabajo seminal de Tukey en 1977 [19] el AED ha ganado muchos seguidores como la metodología de referencia para analizar un conjunto de datos. Según Howard Seltman, doctor en estadística por la Carnegie Mellon University, «en términos generales, cualquier método de análisis de datos que no incluya la modelización y la inferencia estadística formal se engloba dentro del término análisis exploratorio de datos» [16].

El AED es un paso fundamental tras la recogida de datos y el preprocesamiento, en el que los datos simplemente se visualizan, se trazan y se manipulan, sin ninguna suposición, para ayudar a evaluar la calidad de los datos y construir modelos [10].

Los objetivos del AED pueden resumirse como sigue:

1. Maximizar el conocimiento de la base de datos/comprender la estructura de la base de datos.
2. Visualizar las relaciones potenciales entre las variables de exposición y de resultado.
3. Detectar valores atípicos y anomalías.
4. Desarrollar modelos parsimoniosos (un modelo predictivo o explicativo que funcione con el menor número posible de variables de exposición) o una selección preliminar de modelos apropiados;
5. Extraer y crear variables relevantes.

Los métodos del AED pueden clasificarse en:

- Métodos gráficos o no gráficos
- Métodos univariantes o multivariantes.

2.2.1. Análisis Exploratorio de Datos No Gráfico

Los métodos no gráficos permiten conocer las características y la distribución de las variables de interés.

Variables Cualitativas

- **Conteo y/o frecuencia de cada categoría**: Consiste en construir una tabla que contenga el recuento y la frecuencia de datos de cada categoría.

Variables Cuantitativas

Los estadísticos muestrales expresan las características de una muestra mediante un conjunto limitado de parámetros. Suelen considerarse estimaciones de los parámetros correspondientes de la población de la que procede la muestra.

- **Medidas de tendencia central:** media aritmética, mediana, moda.
- **Medidas de dispersión:** varianza, desviación típica, rango intercuartílico, valor máximo y mínimo.
- **Forma de la distribución:** asimetría y curtosis.

Comprobación de la Distribución

Existen varios métodos no gráficos para evaluar la normalidad de un conjunto de datos, como la prueba de Shapiro-Wilk, por ejemplo.

Detección de Valores Atípicos

Varios métodos estadísticos para la detección de valores atípicos entran dentro de las técnicas de AED, como el método de Tukey, la puntuación Z, los residuales studentizados, etc.

Covarianza y Correlación

La covarianza y la correlación miden el grado de relación entre dos variables aleatorias. Una covarianza positiva significa que las variables están positivamente relacionadas, mientras que una covarianza negativa significa que las variables están inversamente relacionadas. Un problema de la covarianza es que su valor depende de la escala de los valores de las variables aleatorias. Cuanto mayores sean los valores de ambas variables, mayor será la covarianza. Esto hace imposible comparar covarianzas de conjuntos de datos con diferentes escalas. Este problema puede solucionarse dividiendo la covarianza por el producto de la desviación típica de cada variable aleatoria, lo que da el coeficiente de correlación de Pearson. La correlación es, por tanto, una versión escalada de la covarianza, que se utiliza para evaluar la relación lineal entre dos variables.

2.2.2. Análisis Exploratorio de Datos Gráfico

Histogramas

Los histogramas se encuentran entre las técnicas de AED más útiles y permiten conocer mejor los datos, incluida la distribución, la tendencia central, la dispersión, la modalidad y los valores atípicos.

Los histogramas son diagramas de barras de recuentos; cada barra representa la frecuencia o la proporción de casos para un intervalo de valores. Los histogramas dan una impresión inmediata de la forma de la distribución (simétrica, uni/multimodal, sesgada, valores atípicos...).

Gráfico de Cajas o *Boxplots*

Los gráficos de caja son una técnica de Análisis Exploratorio de Datos (AED) excelente, ya que permiten representar información sobre la tendencia central, la simetría, la asimetría y los valores atípicos. Sin embargo, pueden ocultar algunos aspectos de los datos, como la multimodalidad [10]. Este gráfico se utiliza para mostrar la distribución de los datos, incluyendo su mediana (Med), el primer cuartil ($Q1$), el tercer cuartil ($Q3$), el rango intercuartílico (IQR), los límites inferior y superior ($[Q1 - k \times IQR,\ Q3 + k \times IQR]$ donde k es una constante que generalmente se toma como 1.5, aunque puede ajustarse según el contexto), y los valores que quedan fuera de estos límites, los cuales se identifican como valores atípicos en el gráfico [2] (Ver Figura 2).

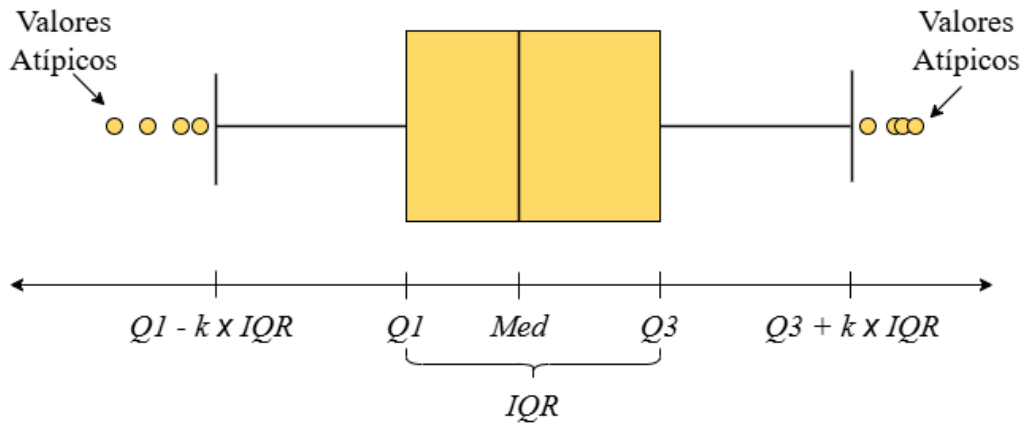

Figura 2: Ejemplo Gráfico de Cajas.

Gráfico de Líneas 2D

Los diagramas de líneas 2D representan gráficamente los valores de una matriz en el eje y, a intervalos regulares en el eje x.

Gráficos de Dispersión

Los gráficos de dispersión se construyen utilizando dos variables cuantitativas continuas, ordinales o discretas. La coordenada de cada punto de datos corresponde a una variable. Pueden complejizarse hasta cinco dimensiones utilizando otras variables, diferenciando el tamaño, la forma o el color de los puntos de datos.

El análisis exploratorio de datos es esencial para cualquier proyecto analítico, ya que permite comprender la estructura y calidad de los datos sin realizar suposiciones previas. Proporciona una base sólida para la selección de modelos y la extracción de variables relevantes, detectando valores atípicos y visualizando relaciones clave entre variables. Mediante métodos gráficos y no gráficos, el AED maximiza el conocimiento de los datos y asegura que los modelos estadísticos y predictivos estén fundamentados en una interpretación precisa y detallada de la información disponible.

2.3. Detección de Valores Atípicos en Series de Tiempo

La detección de valores atípicos se ha estudiado en diversos ámbitos de aplicación, como la detección de fraudes con tarjetas de crédito, la detección de intrusiones en ciberseguridad o el diagnóstico de fallos en la industria. En particular, el análisis de valores atípicos en datos de series temporales examina comportamientos anómalos a lo largo del tiempo [4]. Desde un punto de vista clásico, D. M. Hawkins [9] ha proporcionado una definición muy utilizada del concepto «outlier»:

«Una observación que se desvía tanto de otras observaciones como para despertar sospechas de que ha sido generada por un mecanismo diferente».

Como se muestra en la Figura 3, los valores atípicos en las series temporales pueden llevar al investigador a emprender dos procesos diferentes. En ocasiones, estas observaciones se han relacionado con ruido, datos erróneos o no deseados, que por sí mismos no son interesantes para el analista [1]. En estos casos, los valores atípicos deben eliminarse o corregirse para mejorar la calidad de los datos y generar un conjunto de datos más limpio que pueda ser utilizado por otros algoritmos de minería de datos [4]. Sin embargo, en los últimos años y, especialmente en el ámbito de las series de tiempo, muchos investigadores se han propuesto detectar y analizar fenómenos inusuales pero interesantes. La detección de fraudes es un ejemplo de ello porque el objetivo principal es detectar y analizar el propio fenómeno atípico. Estas observaciones suelen denominarse anomalías [1].

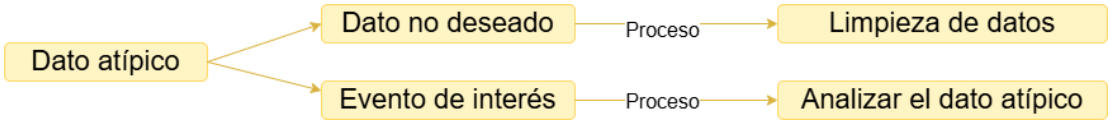

Figura 3: Procesos para lidiar con un dato atípico.

Las técnicas de detección de valores atípicos en series de tiempo varían en función del tipo de dato de entrada, el tipo de dato atípico y la naturaleza del método. Ane Blázquez-García, Ángel Conde, Usue Mori y José A. Lozano [4] propusieron una taxonomía exhaustiva que engloba estos tres aspectos. La Figura 4 muestra una visión general de la taxonomía resultante. A continuación, se presentan algunos conceptos generales, con un énfasis especial en cada uno de los ejes fundamentales de este trabajo.

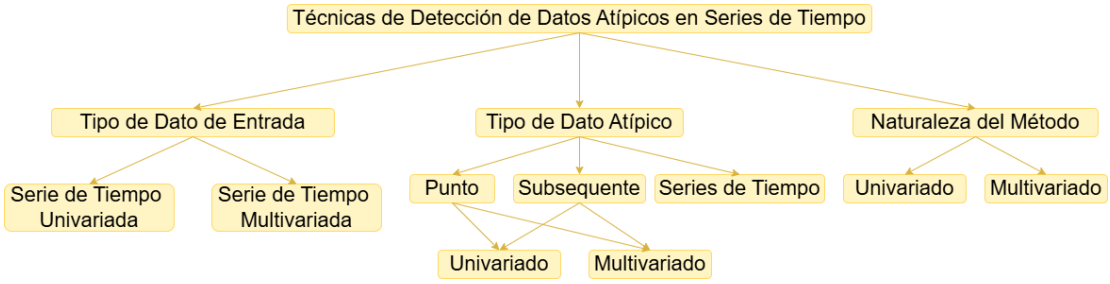

Figura 4: Técnicas de Detección de Datos Atípicos.

2.3.1. Tipo de Dato de Entrada

El primer eje hace referencia al tipo de dato de entrada (Figura 4) que el método de detección es capaz de tratar (es decir, una serie temporal univariante o multivariante).

Definición 2.3.1. (Serie de Tiempo Univariada)

Una serie de tiempo univariada $Y = \{y_t\}_{t \in T}$ es un conjunto ordenado de observaciones de valor real, en el que cada observación se registra en un momento determinado $t \in T \subseteq \mathbb{Z}^+$.

Entonces, y_t es el punto u observación en el momento t y $S = y_p, y_{p+1}, ..., y_{p+n-1}$ la subsecuencia de longitud $n \leq |T|$ a partir de la posición p de la serie temporal Y, para $p, t \in T$ y $p \leq |T| - n + 1$. Se supone que cada observación y_t es un valor observado de una determinada variable aleatoria Y_t.

Definición 2.3.2. (Serie de Tiempo Multivariada)

Una serie de tiempo multivariada $\mathbf{Y} = \{\mathbf{y}_t\}_{t \in T}$ se define como un conjunto ordenado de vectores k-dimensionales, cada uno de los cuales se registra en un momento específico $t \in T \subseteq \mathbb{Z}^+$ y consta de k observaciones de valor real $\mathbf{Y}_t = (y_{1t}, ..., y_{kt})$.

Entonces, se dice que $\mathbf{y}_t$ es un punto y $\mathbf{S} = \mathbf{y}_p, \mathbf{y}_{p+1}, ..., \mathbf{y}_{p+n-1}$ es una subsecuencia de longitud $n \leq |T|$ de una serie de tiempo multivariada $\mathbf{Y}$, para $p, t \in T$ y $p \leq |T| - n + 1$. Para cada dimensión $j \in \{1, ..., k\}$, $Y_j = \{y_{jt}\}_{t \in T}$ es una serie de tiempo univariada y cada observación y_{jt} en el vector $\mathbf{y}_t$ es un valor observado de una variable aleatoria dependiente del tiempo Y_{jt} en $\mathbf{Y}_t = (Y_{1t}, ..., Y_{kt})$. En este caso, cada variable podría depender no sólo de sus valores pasados, sino también de las demás variables dependientes del tiempo.

2.3.2. Tipo de Dato Atípico

El segundo eje (Figura 4) describe el tipo de dato atípico que el método pretende detectar (es decir, un punto, una subsecuencia o una serie temporal).

- **Valores Atípicos Puntuales:** Un valor atípico puntual es un dato que se comporta de forma inusual en un instante de tiempo específico en comparación con los demás valores de la serie temporal (valor atípico global) o con sus puntos vecinos (valor atípico local). Los valores atípicos puntuales pueden ser univariantes o multivariantes, dependiendo de si afectan a una o varias variables dependientes del tiempo, respectivamente.
- **Valores Atípicos Consecutivos:** Este término se refiere a puntos consecutivos en el tiempo cuyo comportamiento conjunto es inusual, aunque cada observación individualmente no es necesariamente un valor atípico puntual. Los valores atípicos subsiguientes también pueden ser globales o locales y pueden afectar a una (valores atípicos subsiguientes univariantes) o más (valores atípicos subsiguientes multivariantes) variables dependientes del tiempo.
- **Series de Tiempo Atípicas:** Las series temporales completas también pueden ser valores atípicos, pero sólo pueden detectarse cuando los datos de entrada son series temporales multivariantes.

Por último, cabe señalar que los valores atípicos dependen del contexto. Así, si el método de detección utiliza toda la serie de tiempo como información contextual, los valores atípicos detectados son globales. Por el contrario, si el método solo utiliza un segmento de la serie (una ventana temporal), los valores atípicos detectados son locales, porque son valores atípicos dentro de su vecindario. Los valores atípicos

globales también son locales, pero no todos los valores atípicos locales son globales. En otras palabras, algunos valores atípicos locales pueden parecer normales si se observa toda la serie de tiempo, pero pueden ser anómalos si nos centramos solo en su vecindario. [4]

2.3.3. Naturaleza del Método

El tercer eje (Figura 4) analiza la naturaleza del método de detección empleado (es decir, si el método de detección es univariante o multivariante). Un método de detección univariante sólo tiene en cuenta una única variable dependiente del tiempo, mientras que un método de detección multivariante puede trabajar simultáneamente con más de una variable dependiente del tiempo. Obsérvese que el método de detección puede ser univariante, aunque los datos de entrada sean series temporales multivariantes, porque se puede realizar un análisis individual de cada variable dependiente del tiempo sin tener en cuenta las dependencias que puedan existir entre las variables. En cambio, no se puede utilizar una técnica multivariante si los datos de entrada son series temporales univariantes.

2.3.4. Técnicas de Detección de Valores Atípicos Puntuales en Series de Tiempo Univariadas

La detección de valores atípicos puntuales es la tarea de detección de valores atípicos más común en el ámbito de las series de tiempo. En esta parte se presentan las técnicas utilizadas para detectar este tipo de valores atípicos en datos de series de tiempo univariadas. En concreto, como se muestra en la Figura 5, a lo largo de este apartado se destacarán dos características clave de estos métodos. En cuanto a la primera característica, el tratamiento de la temporalidad, algunos métodos consideran el orden temporal inherente a las observaciones, mientras que otros ignoran por completo esta información. La principal diferencia entre los métodos que incluyen información temporal y los que no lo hacen es que estos últimos producen los mismos resultados, aunque se apliquen a una versión barajada de la serie.

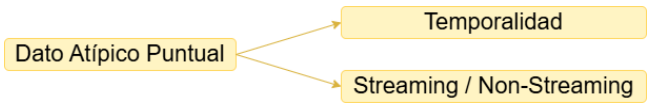

Figura 5: Características Relacionadas con la Detección de Valores Atípicos Puntuales.

En relación con la segunda característica (véase la Figura 5), algunas técnicas son capaces de detectar valores atípicos en series de tiempo de flujo continuo (*streaming*) determinando si un nuevo dato entrante es o no un valor atípico en cuanto llega, sin tener que esperar a recibir más datos. Dentro de este grupo, algunos métodos utilizan un modelo fijo a lo largo de la evolución del flujo, mientras que otros actualizan los modelos con la nueva información recibida, ya sea volviendo a entrenar todo el modelo o aprendiendo de forma incremental.

La definición más popular e intuitiva del concepto de dato atípico puntual es un punto que se desvía significativamente de su valor esperado. Por lo tanto, dada una serie de tiempo univariada, un punto en el tiempo t puede declararse un valor atípico si la distancia a su valor esperado es superior a un umbral predefinido τ :

$$|y_t - \hat{y}_t| > \tau \tag{1}$$

donde y_t es el valor observado y $\hat{y}_t$ es el valor esperado para el tiempo t.

Los métodos de detección de valores atípicos basados en la estrategia descrita en la ecuación (1) se denominan métodos basados en modelos [4]. Aunque cada técnica calcula el valor esperado $\hat{y}_t$ y el umbral τ de forma diferente, todas se basan en el ajuste de un modelo. Como se muestra en la Tabla 1, si $\hat{y}_t$ se obtiene utilizando observaciones anteriores y posteriores a y_t, entonces la técnica se encuentra dentro de los métodos basados en modelos de estimación. Por el contrario, si $\hat{y}_t$ se obtiene basándose únicamente en observaciones anteriores a y_t, entonces la técnica se encuentra dentro de los métodos basados en modelos de predicción. En la práctica, la principal diferencia entre utilizar métodos de estimación o de predicción es que las técnicas dentro de esta última categoría pueden utilizarse en series de tiempo de flujo continuo porque pueden determinar si un nuevo dato es o no un valor atípico en cuanto llega.

Método de Detección	**Datos utilizados**		**Valor esperado**		**Valores atípicos puntuales**
Modelos de estimación	$\{y_{t-k_1}, \ldots, y_t, \ldots, y_{t+k_2}\}$	$\rightarrow$	$\hat{y}_t$	$\rightarrow$	$\lvert y_t - \hat{y}_t \rvert > \tau$
Modelos de predicción	$\{y_{t-k}, \ldots, y_{t-1}\}$	$\rightarrow$	$\hat{y}_t$	$\rightarrow$	

Tabla 1: Estructura de datos utilizada según el método de detección donde $k_1, k_2 \geq 0$ tal que $k_1 + k_2 > 0$ y $k \geq 1$.

Los modelos de estimación más sencillos se basan en modelos constantes o constantes a trozos, en los que se utilizan estadísticos básicos como la mediana o la desviación absoluta de la mediana (MAD) para obtener el valor esperado $\hat{y}_t$. Estos estadísticos se calculan utilizando la serie completa o agrupando los datos en segmentos de igual longitud.

A diferencia de los modelos de estimación, las técnicas basadas en modelos de predicción ajustan un modelo a la serie de tiempo y obtienen $\hat{y}_t$ utilizando únicamente datos pasados; es decir, sin utilizar el punto actual y_t ni ninguna observación posterior. Los puntos que difieren mucho de sus valores predichos se identifican como valores atípicos. Estos modelos son ideales para series de tiempo de flujo continuo. Dentro de los métodos basados en modelos de predicción, algunos utilizan un modelo fijo y, por tanto, no son capaces de adaptarse a los cambios que se producen en los datos a lo largo del tiempo. Otras técnicas se adaptan a la evolución de las series de tiempo reentrenando el modelo periódicamente, o cada vez que llega un nuevo punto. Por tanto, pueden adaptarse a la evolución de los datos. Como enfoque más básico, Basu y Meckesheimer [3] describen un método que predice el valor $\hat{y}_t$ con la mediana de sus datos pasados.

2.4. Método de Tukey para Detectar Datos Atípicos (Rango Intercuartílico)

El método de Tukey [19], también conocido como el método del rango intercuartílico (*IQR*), es una técnica estadística utilizada para identificar valores atípicos en un conjunto de datos. Este método se basa en el cálculo de los cuartiles y el rango intercuartílico para establecer límites que permitan detectar observaciones que se desvían significativamente del resto de los datos.

Procedimiento: Dado un conjunto de datos ordenado $Y = \{y_1, y_2, \ldots, y_n\}$, el método de Tukey sigue los siguientes pasos:

1. **Calcular el primer cuartil (Q_1) y el tercer cuartil (Q_3)**:
   - Q_1 es el valor que deja el 25 % de los datos por debajo de él.
   - Q_3 es el valor que deja el 75 % de los datos por debajo de él.

2. **Calcular el rango intercuartílico** (IQR):

$$IQR = Q_3 - Q_1$$

3. **Establecer los límites para detectar valores atípicos**:

   - **Límite inferior**: $Q_1 - k \times \text{IQR}$
   - **Límite superior**: $Q_3 + k \times \text{IQR}$

   donde k es un factor que generalmente se toma como 1.5, aunque puede ajustarse según el contexto.

4. **Identificar los valores atípicos**:

   - Cualquier valor y_i que sea menor que el límite inferior o mayor que el límite superior se considera un valor atípico.

Este procedimiento se utiliza en conjunto con su respectiva herramienta gráfica (Ver Subsección 2.2.2: Gráfico de Cajas.).

El método de Tukey funciona bien cuando los datos tienen una distribución normal. Sin embargo, el método puede fallar cuando el tamaño de la muestra es pequeño [2]. Los valores atípicos detectados mediante el método de Tukey pueden ser indicativos de errores en la medición, variabilidad natural en los datos, o la presencia de observaciones genuinamente excepcionales. Es importante investigar la naturaleza de estos valores atípicos antes de decidir si deben ser excluidos o tratados de alguna manera en el análisis.

2.5. Desviación Absoluta Mediana

Otra medida de dispersión, que desempeña un papel importante cuando se intenta detectar valores atípicos, es la estadística desviación absoluta mediana [21]. La desviación absoluta mediana (MAD, por sus siglas en inglés) es una medida resistente a la variabilidad, ya que se basa en la mediana como estimación del centro de la distribución, y en la diferencia absoluta en lugar de en la diferencia al cuadrado. [...] Dado que la MAD es la mediana de las desviaciones de las puntuaciones con respecto a la mediana global, no todas las observaciones tienen el mismo peso en esta medida de dispersión. La clara ventaja de la MAD es que evita la influencia de los valores atípicos. Sin embargo, tiene sus propios problemas: si la distribución es realmente normal, se produce una pérdida de eficacia en la medida en que no se aprovecha toda la información disponible en los datos [14].

Definición 2.5.1. (Desviación Absoluta Mediana)

Dado un conjunto de datos $Y = \{y_1, y_2, \ldots, y_n\}$, la Desviación Absoluta Mediana se define como:

$$\text{MAD} = \text{mediana}\left(\{d_i = |y_i - \text{mediana}(Y)| : i = 1, \ldots, n\}\right)$$

donde:

- $\text{mediana}(Y)$ es la mediana del conjunto de datos Y.
- $|y_i - \text{mediana}(Y)|$ es la desviación absoluta del i-ésimo punto con respecto a la mediana.
- $\text{mediana}\left(\{|y_i - \text{mediana}(Y)| : i = 1, \ldots, n\}\right)$ es la mediana de estas desviaciones absolutas.

Para identificar valores atípicos utilizando la MAD, se siguen los siguientes pasos:

1. Calcular la mediana del conjunto de datos:

$$\text{mediana}(Y)$$

2. Calcular las desviaciones absolutas de cada punto x_i con respecto a la mediana:

$$d_i = |y_i - \text{mediana}(Y)|$$

3. Calcular la MAD como la mediana de estas desviaciones absolutas:

$$\text{MAD} = \text{mediana}(\{d_i : i = 1, \ldots, n\})$$

4. Establecer un umbral para detectar valores atípicos. Un enfoque común es considerar como valores atípicos aquellos puntos que se desvían más de $k \times \text{MAD}$ de la mediana, donde k es un factor de escala (típicamente $k = 2.5$ o $k = 3$):

$$\text{Límite inferior} = \text{mediana}(Y) - k \times \text{MAD}$$

$$\text{Límite superior} = \text{mediana}(Y) + k \times \text{MAD}$$

5. Identificar los valores atípicos:
   - Cualquier valor y_i que sea menor que el límite inferior o mayor que el límite superior se considera un valor atípico.

La MAD es especialmente útil en conjuntos de datos con distribuciones no normales.

2.6. PROPHET

La previsión es una tarea común de la ciencia de datos que ayuda a las organizaciones en la planificación de la capacidad, el establecimiento de objetivos y la detección de anomalías [18]. PROPHET es un modelo de series temporales desarrollado por Meta, diseñado para predecir datos con componentes estacionales, tendencias y días festivos de manera intuitiva y robusta. Este modelo se basa en el ajuste de curvas logarítmicas para capturar las tendencias de largo plazo y modelar estacionalidades de manera aditiva o multiplicativa [17].

PROPHET utiliza un modelo de series de tiempo univariadas descomponible [7] con tres componentes principales: tendencia, estacionalidad y días festivos que se combinan en la siguiente ecuación:

$$y_t = y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{2}$$

donde

- $g(t)$ es la función de tendencia que modela los cambios no periódicos en el valor de la serie de tiempo.
- $s(t)$ representa los cambios periódicos (por ejemplo, la estacionalidad semanal y anual).

- $h(t)$ representa los efectos de los días festivos que se producen en horarios potencialmente irregulares durante uno o más días.
- ε_t representa el término de error, es decir, cualquier cambio idiosincrático que no tenga cabida en el modelo; teóricamente se supone que ε_t sigue una distribución normal.
- $t \in T \subseteq \mathbb{Z}^+$.

Las especificaciones de la Ecuación (2) guardan similitud con las de un Modelo Aditivo Generalizado (GAM, por sus siglas en inglés) [8], una familia de modelos de regresión que emplea suavizadores potencialmente no lineales sobre los regresores. PROPHET utiliza exclusivamente el tiempo como regresor, pero incorpora múltiples funciones lineales y no lineales del tiempo como componentes del modelo. La representación de la estacionalidad como un término aditivo sigue el mismo enfoque adoptado en el suavizado exponencial [6]. No obstante, la estacionalidad también puede ser de naturaleza multiplicativa, en cuyo caso su efecto se expresa como un factor que multiplica a $g(t)$.

Una de las principales ventajas de la formulación GAM es su flexibilidad para descomponerse en distintos componentes y admitir nuevas fuentes de estacionalidad cuando estas son identificadas. Además, los GAM permiten un ajuste rápido, lo que facilita la modificación interactiva de los parámetros del modelo en función de las necesidades del usuario.

Desde un punto de vista teórico, PROPHET aborda la predicción como un problema de ajuste de curvas, lo que lo distingue de los modelos clásicos de series temporales que consideran explícitamente la estructura de dependencia temporal en los datos. Este enfoque implica renunciar a ciertas ventajas inferenciales que ofrecen modelos generativos como ARIMA.

El diseño de PROPHET responde tanto a la naturaleza de las series temporales analizadas en Facebook (caracterizadas por tendencias a tramos y múltiples patrones estacionales) como a los desafíos que plantea la previsión a gran escala.

2.7. Métricas de Error en Series de Tiempo

Para evaluar la precisión de los modelos de predicción en series temporales, se emplean diversas métricas de error. A continuación, se presentan las definiciones teóricas de las métricas utilizadas en este estudio, junto con sus respectivas ecuaciones naturales y ecuaciones recurrentes derivadas. Estas últimas fueron empleadas en el análisis para evaluar continuamente las predicciones en relación con los valores reales que se registran diariamente.

2.7.1. Raíz del Error Cuadrático Medio (RMSE)

La RMSE mide la magnitud promedio de los errores en una predicción, penalizando en mayor medida los errores grandes debido a la elevación al cuadrado. Se define como:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{3}$$

donde y_t representa el valor real en el instante t, $\hat{y}_t$ es el valor predicho y n es el número total de observaciones.

Con un par de manipulaciones algebraicas, la Ecuación 3 puede escribirse como sigue (Ecuación 4):

$$\text{RMSE}_t = \sqrt{\frac{(t-1)\cdot \text{RMSE}_{t-1}^2 + (y_t - \hat{y}_t)^2}{t}} \tag{4}$$

donde $t \in T \subseteq \mathbb{Z}^+$ y $\text{RMSE}_0 = 0$.

2.7.2. Error Absoluto Medio (MAE)

El MAE calcula el promedio de los errores absolutos entre las predicciones y los valores reales. A diferencia del RMSE, el MAE no otorga mayor peso a los errores grandes, proporcionando así una medida más interpretable de la desviación media. Se recomienda cuando los valores atípicos representan partes corruptas de los datos [5]. Se expresa como:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |y_t - \hat{y}_t| \tag{5}$$

Con un par de manipulaciones algebraicas, la Ecuación 5 puede escribirse como sigue (Ecuación 6):

$$\text{MAE}_t = \frac{(t-1)\cdot \text{MAE}_{t-1} + |y_t - \hat{y}_t|}{t} \tag{6}$$

donde $t \in T \subseteq \mathbb{Z}^+$ y $\text{MAE}_0 = 0$.

2.7.3. Error Absoluto Porcentual Medio (MAPE)

El MAPE representa el error absoluto medio como un porcentaje del valor real, permitiendo comparar el desempeño del modelo en distintas escalas. Es útil cuando las variaciones relativas son más importantes que las absolutas. Sin embargo, solo se puede usar en datos estrictamente positivos y está sesgado hacia pronósticos bajos [5]. Su formulación es:

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right| \tag{7}$$

Con un par de manipulaciones algebraicas, la Ecuación 7 puede escribirse como sigue (Ecuación 8):

$$\text{MAPE}_t = \frac{(t-1)\cdot \text{MAPE}_{t-1} + \frac{|y_t - \hat{y}_t|}{|y_t|} \times 100}{t} \tag{8}$$

donde $t \in T \subseteq \mathbb{Z}^+$ y $\text{MAPE}_0 = 0$.

Cabe resaltar que el MAPE puede verse afectado por valores de y_t cercanos a cero, lo que puede generar valores excesivamente altos o indefinidos.

2.7.4. Error Absoluto Porcentual Medio Simétrico (sMAPE)

El sMAPE es una variante del MAPE diseñada para corregir problemas de asimetría cuando los valores reales son cercanos a cero. Se define como:

$$\text{sMAPE} = \frac{100}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{\frac{|y_t| + |\hat{y}_t|}{2}} \tag{9}$$

Con un par de manipulaciones algebraicas, la Ecuación 9 puede escribirse como sigue (Ecuación 10):

$$\text{sMAPE}_t = \frac{(t-1) \cdot \text{sMAPE}_{t-1} + \frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2} \times 100}{t} \tag{10}$$

donde $t \in T \subseteq \mathbb{Z}^+$ y $\text{sMAPE}_0 = 0$.

A diferencia del MAPE, el sMAPE emplea el denominador promedio entre el valor real y el predicho, asegurando una mejor estabilidad en casos de valores pequeños de y_t.

3. Metodología

El presente trabajo se llevó a cabo en el área de Tecnologías de la Información (TI) del Grupo Éxito, específicamente dentro del equipo de Gestión de MIPS. Para la gestión del trabajo se utilizó el marco metodológico Scrum, el cual permitió una organización iterativa e incremental de las tareas. Se eligió esta metodología debido a la falta de claridad inicial sobre los requisitos funcionales y no funcionales del microservicio. Dado que Scrum se basa en un enfoque empírico y promueve la mejora continua a través de cada *Sprint*, se consideró la opción más adecuada para abordar un problema con una solución aún indefinida. Además, la flexibilidad del marco permitió la construcción de un Trabajo Pendiente del Microservicio (*Product Backlog*) adaptable, facilitando remover o incorporar nuevos ajustes a medida que avanzaba el desarrollo.

3.1. Planteamiento del Problema

Para ofrecer un producto, servicio o solución debe existir una demanda, una necesidad, un problema. En este caso, la problemática surgió a raíz del alto costo asociado al consumo de MIPS en la plataforma ClearPath. En febrero de 2021, Grupo Éxito firmó un contrato con Unisys—proveedor del servicio ClearPath—para el uso de 281,400 MIPS, con un valor cercano a los 27 millones de dólares. Este contrato tiene como fecha de finalización septiembre de 2027, con la posibilidad de extenderse por seis meses adicionales en caso de que los recursos habilitados no se hayan agotado para entonces. Debido al impacto financiero de este recurso, la optimización del consumo de MIPS se convirtió en una prioridad estratégica para generar ahorros en los costos operacionales de la empresa.

Sin embargo, el Equipo de Gestión de MIPS enfrentaba limitaciones significativas en sus herramientas de monitoreo y análisis. Una de sus tareas diarias era identificar los procesos que, en un día determinado, habían consumido una cantidad de MIPS significativamente superior a su valor esperado. Este análisis, realizado de forma manual, era repetitivo, laborioso y lento, lo que retrasaba la comunicación con el Equipo de Operaciones, responsable de intervenir y detener estos procesos con consumos anómalos. Además, dado que diariamente se ejecutaban alrededor de 2,500 procesos, se hacía evidente la necesidad de aprovechar herramientas tecnológicas que permitieran manejar grandes volúmenes de datos con mayor eficiencia.

Para abordar esta necesidad, se habían implementado tableros de monitoreo en *Power BI* y *Microstrategy*, los cuales proporcionaban visualizaciones del consumo de MIPS. No obstante, estas plataformas presentaban deficiencias que limitaban su utilidad. En primer lugar, los tableros no contaban con mecanismos avanzados de detección automática de anomalías, lo que obligaba a los analistas a realizar evaluaciones manuales para identificar patrones inusuales. Adicionalmente, la problemática no solo radicaba en el monitoreo y detección de consumos atípicos, sino también en la proyección futura del consumo de MIPS. Era fundamental desarrollar un modelo capaz de identificar tendencias y patrones históricos para prever el agotamiento de los recursos habilitados e integrar los resultados del modelo en los tableros de *Power BI* para que fueran fácilmente interpretados. La Figura 6 ilustra la problemática expuesta.

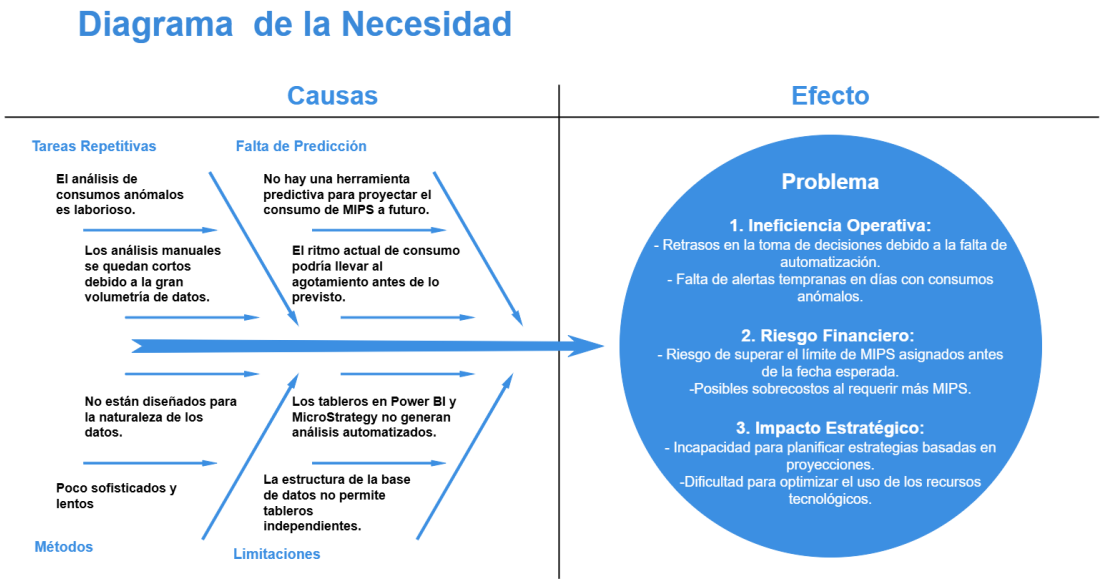

Figura 6: Diagrama Causa-Efecto de la Necesidad.

3.2. Objetivo del Microservicio

Se planteó el objetivo de desarrollar un microservicio que implementara dos modelos analíticos para generar reportes diarios de forma automática. El primer modelo estaría enfocado en detectar procesos con consumos de MIPS anómalos, aplicando la metodología del rango intercuartílico cuando los consumos, agrupados por proceso y día de la semana, presentaban una distribución normal, y la metodología de la desviación absoluta mediana en caso contrario. El segundo modelo, por su parte, estaría diseñado para predecir el consumo total de MIPS, asegurando que las métricas de error—RMSE, MAE, MAPE y sMAPE—se mantuvieran por debajo del 15 %. Finalmente, toda la información generada debía integrarse en los tableros de Power BI habilitados por los sistemas productivos de la empresa, asegurando una visualización accesible y efectiva para todos los actores involucrados en la toma de decisiones.

3.3. Trabajo Pendiente (*Backlog*) del Microservicio

La Tabla 2 presenta los elementos priorizados para el desarrollo del microservicio de monitoreo y optimización del consumo de MIPS. Su estructura responde a la necesidad de automatizar el análisis, mejorar la eficiencia en la detección de consumos atípicos, generar predicciones estratégicas y visualizar la información a través de un tablero de *Power BI*.

ID	Tarea	Prioridad
1	Establecer requerimientos funcionales y no funcionales del microservicio.	Alta
2	Identificar las tablas que contenían la información relevante del consumo de MIPS de las bases de datos `WLMDW` y `WLMDW_TEMPO`.	Alta
3	Realizar un análisis exploratorio de datos sobre la tablas relevantes halladas.	Alta
4	Diseñar la estructura de una base de datos normalizada para contener la información de las tablas relevantes y las nuevas funcionalidades.	Alta
5	Diseñar la arquitectura del microservicio reflejando el flujo de trabajo del mismo y la conexión con las diferentes aplicaciones y *softwares* de la compañía.	Media
6	Construir el código del modelo detector de datos atípicos usando el lenguaje de programación *Python*.	Alta
7	Construir el código del modelo predictivo usando *Python* y *PROPHET*.	Alta
8	Escribir el código que construya las tablas, en una nueva base de datos llamada `Consumo-PrediccionesMIPS`, tomando en cuenta la estructura normalizada diseñada.	Alta
9	Construir el código que integre los modelos y la manipulación de los datos para hacer extracciones e inserciones en las respectivas tablas y bases de datos.	Alta
10	Construir un tablero de monitoreo usando *Power BI Desktop* y conexiones ODBC a la base de datos normalizada, asegurando que la información reflejada en el tablero es consistente y verificable.	Alta
11	Construir el *pipeline*, el *cronjob* (usando YAML) y el *Dockerfile* necesarios para generar una integración consistente en los entornos productivos.	Alta
12	Hacer pruebas y validaciones de la integración del microservicio en el entorno o rama de desarrollo.	Alta
13	Desarrollar la documentación completa del repositorio que contiene el microservicio.	Media
14	Generar un *pull request* al entorno productivo o rama *master* garantizando una ejecución exitosa del microservicio en el OKS (*On-Premise Kubernetes*).	Alta
15	Subir el tablero construido con *Power BI Desktop* a la aplicación productiva de *Power BI de Microsoft 365*, habilitando la visualización de datos a los interesados.	Alta
16	Generar la documentación del proyecto.	Media

Tabla 2: Trabajo Pendiente del Microservicio.

El Trabajo Pendiente del Microservicio presentado en la Tabla 2 es un documento dinámico que se adaptó en función del avance del proyecto y de la retroalimentación recibida durante los *Sprints*.

3.4. *Sprints*

El desarrollo del microservicio se llevó a cabo en un período de aproximadamente seis meses, iniciando el 22 de julio de 2024 y finalizando el 17 de enero de 2025. Para una gestión eficiente del trabajo, el proyecto se organizó en seis *Sprints*, cada uno con una duración de un mes, salvo el último.

3.4.1. *Sprint* 1

Entre el 22 de julio y el 21 de agosto de 2024, se llevó a cabo el primer *Sprint* con el propósito de sentar las bases para el desarrollo del microservicio de monitoreo de MIPS. Dado que en esta etapa inicial aún no se tenía una visión clara de la solución final ni de los recursos disponibles para su implementación, resultaba fundamental realizar una exploración exhaustiva de los datos almacenados en las bases de datos y comprender el comportamiento del consumo de MIPS. Para ello, se extrajeron y analizaron datos de las tablas más relevantes de las bases de datos `WLMDW` y `WLMDW_TEMPO` en *SQL Server* utilizando conexiones ODBC desde *Python*. El análisis exploratorio permitió identificar medidas de tendencia central, medidas de variabilidad, distribuciones y naturaleza de los datos; se tuvo en especial consideración la agrupación de los consumos respecto a las variables de tipo texto.

En paralelo, se definieron los requerimientos funcionales y no funcionales del microservicio, considerando las historias de usuario proporcionadas por el *Equipo de Gestión de MIPS* y el *Equipo de Operaciones*, así como las necesidades identificadas en el Planteamiento del Problema (Ver Sección 3.1). La recopilación y análisis de estos requisitos garantizaron una base sólida para las fases posteriores del desarrollo. En la Tabla 3 se pueden observar los elementos del Trabajo Pendiente del Microservicio seleccionados para este *Sprint*.

ID	Tarea	Prioridad
1	Establecer requerimientos funcionales y no funcionales del microservicio.	Alta
2	Identificar las tablas que contenían la información relevante del consumo de MIPS de las bases de datos `WLMDW` y `WLMDW_TEMPO`.	Alta
3	Realizar un análisis exploratorio de datos sobre la tablas relevantes halladas.	Alta

Tabla 3: Elementos del Trabajo Pendiente del Microservicio Seleccionados para el *Sprint* 1.

***Objetivo del Sprint*:** Obtener una visión general de los datos y establecer los requerimientos funcionales y no funcionales del microservicio.

3.4.2. *Sprint* 2

El segundo *Sprint* tuvo lugar entre el 22 de agosto de 2024 y el 21 de septiembre de 2024. El objetivo de este segundo *Sprint* surgió como consecuencia del primero. Se determinó que una base de datos normalizada agilizaría la carga de datos y facilitaría la construcción de los tableros en *Power BI*.

Por lo tanto, el propósito de este *Sprint* fue diseñar la estructura de una base de datos normalizada y la arquitectura del flujo de trabajo del microservicio haciendo uso del *software* de dibujo de gráficos *draw io*. Además, se esperaba iniciar el desarrollo del código del modelo detector de anomalías haciendo uso

de *Python* y tomando en cuenta los hallazgos obtenidos en el análisis exploratorio de datos realizado en el *Sprint* anterior.

ID	Tarea	Prioridad
4	Diseñar la estructura de una base de datos normalizada para contener la información de las tablas relevantes y las nuevas funcionalidades.	Alta
5	Diseñar la arquitectura del microservicio reflejando el flujo de trabajo del mismo y la conexión con las diferentes aplicaciones y *softwares* de la compañía.	Media
6	Construir el código del modelo detector de datos atípicos usando el lenguaje de programación *Python*.	Alta

Tabla 4: Elementos del Trabajo Pendiente del Microservicio Seleccionados para el *Sprint* 2.

***Objetivo del Sprint*:** Diseñar la arquitectura de la base de datos de destino, el flujo de trabajo del microservicio y construir el módulo con el código para catalogar datos atípicos.

3.4.3. *Sprint* 3

Entre el 22 de septiembre y el 21 de octubre de 2024, el tercer *Sprint* se centró en el desarrollo del modelo predictivo y la consolidación de la estructura de datos diseñada previamente. Como primer paso, se implementó el modelo predictivo utilizando *Python* y la librería *PROPHET*, con el objetivo de anticipar el consumo futuro de MIPS a partir de los patrones históricos identificados. En paralelo, se construyó el código en *Python* que crea las tablas en la base de datos de destino de datos llamada `Consumo-PrediccionesMIPS`, asegurando que su diseño siguiera la estructura normalizada establecida en el *Sprint* 2. Finalmente, se desarrolló el código necesario para integrar los modelos y gestionar la manipulación de datos, permitiendo la extracción y posterior inserción en la base de datos correspondiente. Este *Sprint* resultó clave para garantizar que el microservicio pudiera no solo detectar anomalías y generar predicciones, sino también gestionar adecuadamente la manipulación de datos para insertar la información en sus correspondientes tablas.

ID	Tarea	Prioridad
7	Construir el código del modelo predictivo usando *Python* y *PROPHET*.	Alta
8	Escribir el código que construya las tablas, en una nueva base de datos llamada `Consumo-PrediccionesMIPS`, tomando en cuenta la estructura normalizada diseñada en el *Sprint* anterior.	Alta
9	Construir el código que integre los modelos y la manipulación de los datos para hacer extracciones e inserciones en las respectivas tablas y bases de datos.	Alta

Tabla 5: Elementos del Trabajo Pendiente del Microservicio Seleccionados para el *Sprint* 3.

***Objetivo del Sprint*:** Construir módulos de código para extraer, transformar y cargar los datos en la base de datos destino.

3.4.4. *Sprint* 4

Entre el 22 de octubre y el 21 de noviembre de 2024, el cuarto *Sprint* se enfocó en la construcción de herramientas clave para la visualización y automatización del microservicio. En primer lugar, se desarrolló un tablero de monitoreo en *Power BI Desktop*, utilizando conexiones *Direct Query* a la base de datos normalizada. Esto permitió garantizar que la información reflejada en los reportes fuera consistente, verificable y actualizada en tiempo real, optimizando la capacidad de análisis del equipo de Gestión de MIPS.

Adicionalmente, se implementaron los componentes necesarios para la integración y despliegue del microservicio en entornos productivos. Para ello, se desarrolló un ***pipeline***, junto con un ***cronjob*** en *YAML* y un ***Dockerfile***, asegurando la ejecución automatizada de los procesos y una transición fluida entre entornos de desarrollo y producción. Este Sprint marcó un avance significativo hacia la implementación operativa del microservicio de forma automática.

ID	**Tarea**	**Prioridad**
10	Construir un tablero de monitoreo usando *Power BI Desktop* y conexiones (*Direct Query*) a la base de datos normalizada, asegurando que la información reflejada en el tablero es consistente y verificable.	Alta
11	Construir el *pipeline*, el *cronjob* (usando YAML) y el *Dockerfile* necesarios para generar una integración consistente en los entornos productivos.	Alta

Tabla 6: Elementos del Trabajo Pendiente del Microservicio Seleccionados para el *Sprint* 4.

***Objetivo del Sprint*:** Construir el tablero que optimizara el análisis de los consumos de MIPS y los archivos necesarios para la integración y despliegue del microservicio en producción.

3.4.5. *Sprint* 5

Entre el 22 de noviembre y el 21 de diciembre de 2024, el quinto *Sprint* se centró en la validación y documentación del microservicio, asegurando su estabilidad antes de su despliegue en producción. Se llevaron a cabo pruebas y validaciones exhaustivas en el entorno de desarrollo, con el objetivo de verificar la correcta integración del microservicio con las bases de datos, modelos de análisis y componentes previamente implementados. Durante esta fase, se identificaron y corrigieron posibles fallos, optimizando el rendimiento y garantizando la fiabilidad del sistema.

Paralelamente, se desarrolló la documentación completa del repositorio del microservicio, detallando su arquitectura, configuración, flujo de datos y procesos de despliegue. Esta documentación no solo facilitó la comprensión del sistema por parte de los desarrolladores y administradores, sino que también estableció una base para futuras mejoras y mantenimientos. Con este Sprint, el microservicio alcanzó un nivel de madurez que permitió avanzar hacia su implementación final en entornos productivos.

ID	Tarea	Prioridad
12	Hacer pruebas y validaciones de la integración del microservicio en el entorno o rama de desarrollo.	Alta
13	Desarrollar la documentación completa del repositorio que contiene el microservicio.	Media

Tabla 7: Elementos del Trabajo Pendiente del Microservicio Seleccionados para el *Sprint* 5.

***Objetivo del Sprint*:** Validar y verificar la correcta integración y despliegue del microservicio en la rama de desarrollo. Generar la documentación del repositorio.

3.4.6. *Sprint* 6

Entre el 22 de diciembre de 2024 y el 17 de enero de 2025, el sexto *Sprint* marcó la fase final del desarrollo del microservicio, enfocándose en su despliegue en producción y la consolidación de su documentación. Se llevó a cabo la generación de un ***pull request*** al entorno productivo o rama *master*, garantizando que la ejecución del microservicio en el *OKS (On-Premise Kubernetes)* fuera exitosa. Esta tarea aseguró que la solución estuviera completamente integrada en la infraestructura tecnológica de la compañía, asegurando su completa ejecución automática.

En paralelo, se publicó el tablero de monitoreo en la aplicación productiva de *Power BI de Microsoft 365*, habilitando la visualización de datos para los interesados y permitiendo un seguimiento efectivo del consumo de MIPS. Finalmente, se completó la documentación del proyecto, consolidando los detalles técnicos, funcionales y operativos del microservicio, lo que facilitaría su mantenimiento y futuras mejoras. Con este *Sprint*, el desarrollo alcanzó su fase de finalización.

ID	Tarea	Prioridad
14	Generar un *pull request* al entorno productivo o rama *master* garantizando una ejecución exitosa del microservicio en el OKS (*On-Premise Kubernetes*).	Alta
15	Subir el tablero construido con *Power BI Desktop* a la aplicación productiva de *Power BI de Microsoft 365*, habilitando la visualización de datos a los interesados.	Alta
16	Generar la documentación del proyecto.	Media

Tabla 8: Elementos del Trabajo Pendiente del Microservicio Seleccionados para el *Sprint* 6.

***Objetivo del Sprint*:** Asegurar la completa integración del microservicio en producción, garantizando su ejecución automática.

Cada *Sprint* permitió la planificación, ejecución y evaluación iterativa de los avances del microservicio (Ver Figura 1), asegurando una evolución continua y alineada con los objetivos del proyecto (Ver Sección 3.2 y Tabla 2).

4. Resultados y Discusión

En esta sección se presentan los resultados de cada *Sprint*, detallando el trabajo realizado, los desafíos enfrentados, los ajustes implementados y los logros alcanzados a lo largo del desarrollo. Se ofrece una visión integral del proceso, resaltando la evolución del proyecto y las decisiones clave que marcaron su progreso.

4.1. Requerimientos funcionales y no funcionales de microservicio

Los requisitos funcionales especifican las capacidades y operaciones que el sistema debe realizar, mientras que los requisitos no funcionales establecen criterios de calidad y restricciones que garantizan su correcto desempeño.

4.1.1. Requisitos Funcionales

- Ejecutarse automáticamente desde el *OKS* de Grupo Éxito todos los días a las 9:30 AM.
- Catalogar los procesos con consumos atípicos de MIPS según el día de la semana.
- Generar predicciones del consumo de MIPS hasta el completo agotamiento de los recursos habilitados.
- El sistema debe proporcionar una interfaz, como un tablero de *Power BI*, que permita a los usuarios agregados en el *cluster* de acceso visualizar los datos procesados.
- Manejar excepciones y cargar únicamente los datos nuevos detectados en la base de datos de origen.

4.1.2. Requisitos No Funcionales

Además, el microservicio debe cumplir con los siguientes estándares de calidad y rendimiento:

- Garantizar un tiempo de ejecución óptimo, asegurando que el procesamiento de datos no exceda las 10:30 AM del mismo día de ejecución.
- Restringir modificaciones no autorizadas en los datos procesados.
- Asegurar una disponibilidad del 99.9 %, minimizando interrupciones en la ejecución programada.
- Ofrecer una estructura escalable que permita adaptaciones futuras sin afectar el rendimiento del sistema.
- Garantizar que las métricas de precisión—RMSE, MAE, MAPE y sMAPE—del modelo predictivo se mantengan por debajo del 15 %.

4.2. Análisis Exploratorio de Datos

Como resultado de la ejecución de la tarea identificada con el ID 2 del Trabajo Pendiente del Microservicio seleccionada para el *Sprint* 1 (ver Tabla 3), se identificó que la tabla `refrescarprocesos_10dias`, perteneciente a la base de datos `WLMDW_TEMPO`, contenía la información más relevante sobre el consumo de MIPS. Dicha tabla estaba compuesta por cinco columnas que registraban datos clave, incluyendo el consumo de MIPS y el número total de ejecuciones, segmentados por nombre del proceso, nombre del grupo y fecha. En la Figura 7 se presenta un encabezado que ilustra la estructura general de la tabla `refrescarprocesos_10dias`, mientras que en la Tabla 9 se proporciona un desglose más detallado de la naturaleza de sus variables. Esta fuente de información constituyó el pilar fundamental para el desarrollo del resto del proyecto.

NombreProceso	NombreGrupo	Fecha	total_ejecucionesFecha	total_mipsFecha
WFL/OPCON/COPIA/SALIDA/DREMPUMODA/INDEXA	EVENTOS	2022-12-30	1	1.4083594584012866e-06
WFL/OPCON/WRBAJADETR	DESPACHOS	2022-12-30	4	4.804070596991056e-06
WFL/OPCON/COPIA/ENTRADA/IRTRAMILAN	OPERACION	2022-12-30	462	0.0009906869883547472
(SINCO)PSINCO/DRAGOTA ON EXF	AGOTADOS	2022-12-30	1	0.005887380692393325
WFL/PSINCO/EJEREPO	OPERACION	2022-12-30	3873	0.00452798519782898
WFL/OPCON/COPIA/SALIDA/CRCOSTEOCR	VENTAS	2022-12-30	1	2.0655938723218873e-06
WFL/OPCON/CREXNITCCL	OPERACION	2022-12-30	1	1.0484453745876245e-06
(SINCO)PSINCO/IRTRANSINT ON EXF	INVENTARIOS	2022-12-30	1	0.000447154128042408...
(SINCO)PSINCO/CRPRIIVA ON EXF	PRECIOS	2022-12-30	1	0.0006407409598188787
(SINCO)PSINCO/WRLIQUIREC ON EXF	LOGISTICA	2022-12-30	81	0.38031019586524456
WFL/OPCON/COPIA/SALIDA/RESPALDO/WRACTFAC...	OPERACION	2022-12-30	1	2.1281876260286107e-06
(SINCO)FTP/FILE/TRANSFER/FROM/SASDDPOPROD	FTP	2022-12-31	1234	0.004347949913730138

Figura 7: Encabezado de la Tabla `dbo.refrescarprocesos_10dias`.

Nombre de la Variable	Tipo	Recuento
NombreProceso	Texto	8120
NombreGrupo	Texto	69
Fecha	Fecha	956
total_ejecucionesFecha	Entero	
total_mipsFecha	Real	

Tabla 9: Estructura de la Tabla `dbo.refrescarprocesos_10dias` y Tipo de Variables.

La tabla `refrescarprocesos_10dias` contiene cinco columnas que registran datos clave sobre el consumo de MIPS y el número total de ejecuciones. A continuación, se proporciona una descripción detallada de cada una de las variables:

- `NombreProceso`: Esta variable almacena el nombre del proceso que se está registrando.
- `NombreGrupo`: Esta variable contiene el nombre del grupo al que pertenece el proceso. Es una variable cualitativa que permite agrupar procesos similares o relacionados.

- `Fecha`: Esta variable registra la fecha en la que se ejecutó el proceso. Es una variable de tipo fecha que permite segmentar y analizar los datos temporalmente.
- `total_ejecucionesFecha`: Esta variable cuantitativa registra el número total de ejecuciones del proceso en la fecha especificada. Indica cuántas veces se ejecutó el proceso en un día determinado.
- `total_mipsFecha`: Esta variable cuantitativa registra el consumo total de MIPS (Million Instructions Per Second) del proceso en la fecha especificada. Indica la cantidad de recursos computacionales utilizados por el proceso en un día determinado. Esta variable es de tipo real no negativa.

La tabla `dbo.refrescarprocesos_10dias` está diseñada para no permitir filas duplicadas, valores nulos ni tipos de datos distintos a los definidos para cada columna. Estas restricciones aseguran la exactitud de los registros, refuerzan la unicidad de los datos y garantizan su completitud. Asimismo, la carga de datos está diseñada para insertar diariamente una cantidad n de registros, correspondiente al número de procesos que estuvieron activos ese día. Por ejemplo, si en la fecha x se ejecutaron n procesos, entonces la tabla almacenará exactamente n registros asociados a dicha fecha, asegurando así una representación precisa de la actividad diaria.

4.2.1. Análisis de las Variables Cualitativas y Tipo Fecha

Como se observa en la Tabla 9, la tabla `dbo.refrescarprocesos_10dias` contiene dos variables de tipo texto, las cuales se clasifican como cualitativas. La variable `NombreProceso` es un elemento clave dentro del conjunto de datos, ya que determina la existencia misma de los registros. Las demás variables, en cambio, cumplen la función de resumir y caracterizar la información asociada a cada proceso.

Como primer paso en el análisis, se examinó la variable `Fecha` con el propósito de determinar el rango temporal de la información. Los registros inician el 1 de enero de 2022 y se actualizan diariamente, dado que cada día se incorporan nuevos datos. Esto implica que, al momento de realizar el análisis, la fecha más reciente registrada correspondía al mes de agosto del año 2024.

Luego, se determinó la cantidad de etiquetas únicas para cada una de las variables cualitativas y la fecha, cuyos resultados se presentan en la Tabla 10.

Nombre de la Variable	NombreProceso	NombreGrupo	Fecha
Nro. Único de Etiquetas	8120	69	956

Tabla 10: Conteo de Etiquetas de las Variables Cualitativas y Variable `Fecha`.

Los resultados presentados en la Tabla 10 evidencian una gran cantidad de procesos existentes en `dbo.refrescarprocesos_10dias`. Al analizar la frecuencia con la que se registra cada proceso a lo largo del historial de datos, se observó un rango que oscila entre un mínimo de 1 registro y un máximo de 2797 registros. Inicialmente, esto generó una discordancia, ya que se esperaba que cada proceso tuviera un único registro por fecha, lo que habría limitado el máximo del rango a la cantidad total de fechas registradas, es decir, 956. Sin embargo, el líder del equipo de Gestión de MIPS aclaró que algunos procesos, aunque comparten el mismo nombre, pertenecen a grupos distintos. Por lo tanto, la unicidad de los registros no se veía comprometida, siempre y cuando se considerara la agrupación por `NombreProceso` y `NombreGrupo`. Este factor es relevante, ya que para identificar correctamente un proceso como único se requiere la agrupación por `NombreProceso` y `NombreGrupo`.

Al aplicar el criterio anterior, los datos mostraron mayor consistencia, reduciendo el rango de frecuencia con la que se registra cada proceso a valores entre 1 y 956. Esto indica que algunos procesos se registraron

una única vez, otros en todas las fechas y el resto en un número de ocasiones intermedio, entre 2 y 955 veces. La Figura 8 muestra la distribución de la frecuencia con la que los procesos únicos fueron registrados a lo largo del periodo de análisis. Se observa una fuerte asimetría, donde la mayoría de los procesos presentan pocas activaciones, mientras que otro subconjunto mantiene una activación constante en todas las fechas disponibles.

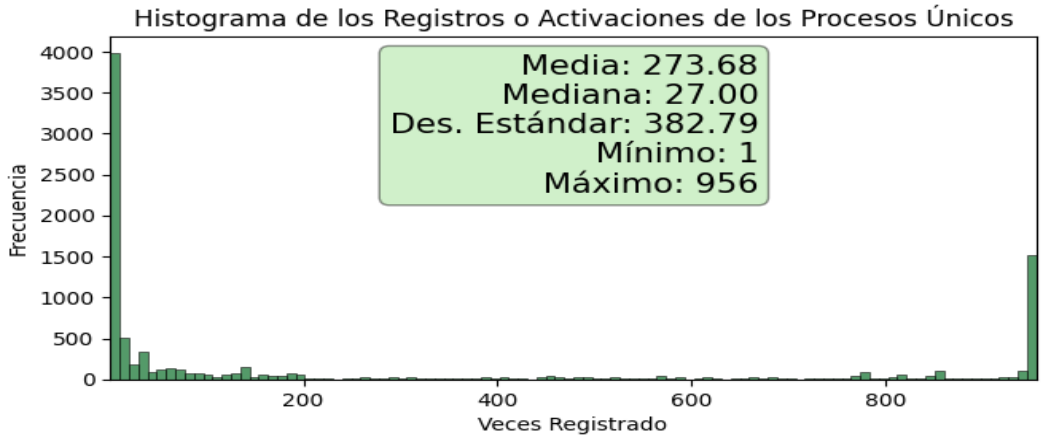

Figura 8: Histograma de frecuencia de los registros o activaciones de los procesos únicos.

La Figura 9 proporciona un análisis detallado de la frecuencia con la que los procesos han sido registrados en la mitad inferior de la distribución (por debajo del percentil 50). Se observa que la mayoría de estos procesos tienen entre 1 y 5 activaciones, lo que sugiere que una parte significativa está relacionada con tareas esporádicas o eventos específicos dentro de la operación del sistema. Además, la distribución presenta una alta concentración en los valores más bajos y una menor variabilidad en comparación con el total de los procesos, lo que indica que esta primera mitad está dominada por procesos de baja recurrencia.

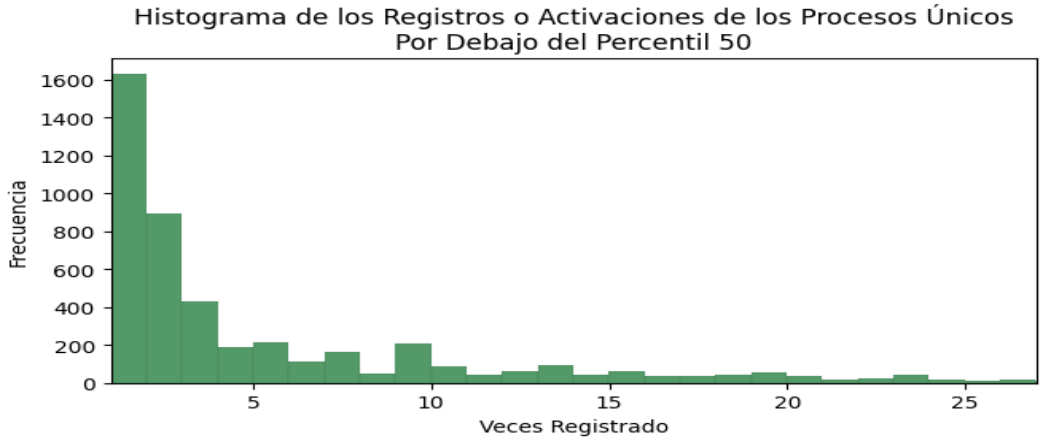

Figura 9: Histograma de frecuencia de los registros o activaciones de los procesos únicos reducido a la mitad ordenada inferior.

La Figura 10 muestra la distribución de la frecuencia con la que los procesos han sido registrados en la mitad superior de la distribución (por encima del percentil 50). Se observa que, a diferencia de la mitad inferior, la mayoría de los procesos en este rango presentan una distribución dispersa con valores de activación significativamente más altos.

La gráfica revela que, aunque existen procesos con activaciones relativamente bajas dentro de esta mitad, hay una acumulación notable en los extremos superiores, con un grupo reducido de procesos que han sido registrados en más de 800 ocasiones. Esto sugiere la existencia de un conjunto de procesos altamente recurrentes dentro del sistema, posiblemente asociados a funciones críticas o esenciales para la operación diaria. Asimismo, se observa que la frecuencia de procesos en la parte central de la distribución es baja en comparación con los valores extremos.

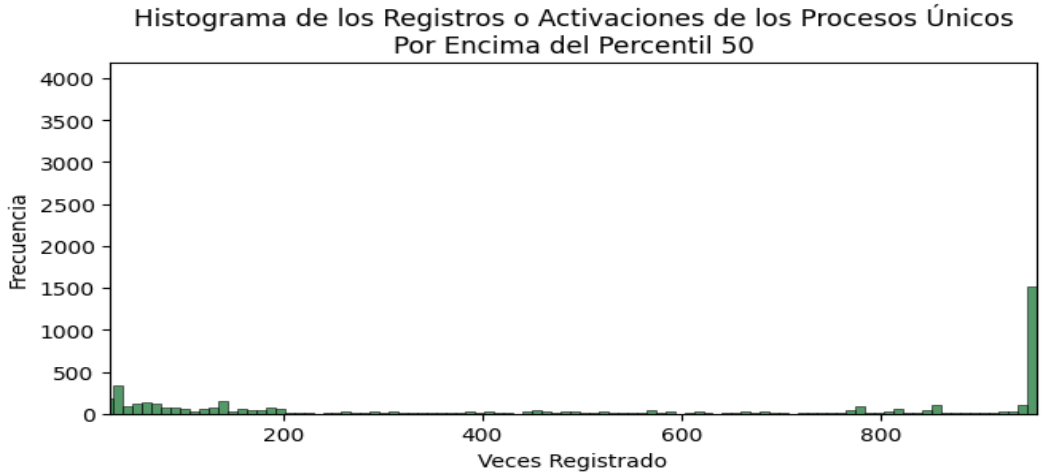

Figura 10: Histograma de frecuencia de los registros o activaciones de los procesos únicos reducido a la mitad ordenada superior.

En conjunto, la distribución de los registros de los procesos indica que la mayoría de los procesos no presentan una activación recurrente, mientras que un segundo grupo se registra de manera constante. Esto podría reflejar una estructura operativa dividida entre procesos críticos y procesos eventuales dentro del sistema analizado. Además, según el líder del equipo, algunos de los procesos con pocas activaciones corresponden a procesos anuales de inventario, los cuales, a pesar de ejecutarse en contadas ocasiones, tienden a consumir una cantidad significativa de recursos debido al alto volumen de información que procesan.

Con el fin de analizar la proporción de recursos consumidos por los diferentes tipos de procesos, se decidió agruparlos en tres segmentos: procesos registrados una sola vez, procesos con un número intermedio de registros (entre 2 y 955 veces) y procesos registrados en todas las fechas. Posteriormente, se evaluó la proporción que ocupa cada segmento (Ver Figura 11) y, finalmente, se identificó la proporción de recursos, en términos de las variables `total_ejecucionesFecha` y `total_mipsFecha`, que cada uno de los segmentos demanda (Ver Figura 12), permitiendo así una mejor comprensión del impacto operativo de cada categoría.

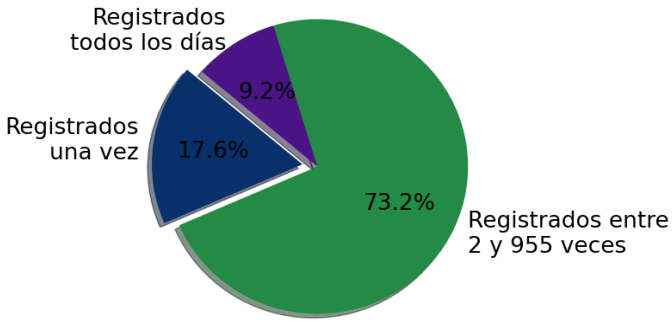

Figura 11: Proporción de registros de los segmentos: procesos registrados una sola vez, procesos con un número intermedio de registros (entre 2 y 955 veces) y procesos registrados en todas las fechas.

En la Figura 11 se identifican tres segmentos principales: procesos registrados una sola vez (17.6 %), procesos registrados en todas las fechas (9.2 %) y procesos con un número de registros intermedio, entre 2 y 955 veces (73.2 %).

La distribución observada revela que la mayoría de los procesos (73.2 %) tienen una frecuencia de registro variable, lo que indica que su activación sigue un patrón intermedio en lugar de ocurrir de forma constante o esporádica. Este comportamiento sugiere que dichos procesos están vinculados a requerimientos operacionales específicos que determinan su activación en distintos días.

Por otra parte, el 17.6 % de los procesos se activaron solo una vez en todo el historial de datos. Este comportamiento podría estar asociado a pruebas, procesos descontinuados o eventos excepcionales que requirieron una activación puntual.

Finalmente, el 9.2 % de los procesos se registraron en todas las fechas disponibles (956 días), lo que sugiere que forman parte de la operación diaria del sistema. Este grupo es especialmente relevante, ya que cualquier interrupción en su activación podría indicar fallos en la infraestructura.

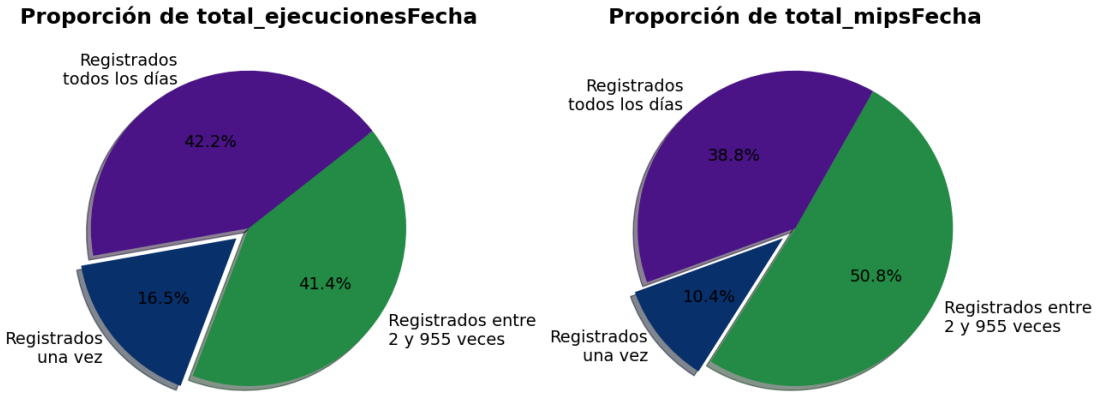

Figura 12: Proporción de ejecuciones y consumo de MIPS por segmento.

La Figura 12 presenta dos diagramas de pastel. El primero (de izquierda a derecha) muestra la propor-

ción de ejecuciones, según la variable `total_ejecucionesFecha`, para cada uno de los segmentos. El segundo diagrama muestra la proporción de consumo de MIPS, según la variable *total_mipsFecha*, para cada uno de los segmentos.

Es importante diferenciar entre los términos activación y ejecución. Una **activación** o **registro** se refiere al número de veces en los que un proceso único ha sido registrado o activado en un conjunto de días. Dado que un proceso solo puede activarse una vez por día, el número de activaciones es equivalente a la cantidad de días en los que estuvo presente. Por otro lado, una **ejecución** representa la cantidad de veces que un proceso único ha sido ejecutado dentro de un período de tiempo determinado. A diferencia de las activaciones, en un solo día pueden registrarse múltiples ejecuciones de un mismo proceso.

Con base en lo anterior, la interpretación del primer diagrama de la Figura 12 es la siguiente: el segmento correspondiente a los procesos registrados solo una vez representa el 16.5 % del total de ejecuciones. Por otro lado, los procesos registrados en todas las fechas constituyen el 42.2 % del total de ejecuciones. Finalmente, los procesos con un número intermedio de registros abarcan el 41.4 % de las ejecuciones totales.

De manera similar, la interpretación del segundo diagrama de la Figura 12 es la siguiente: el segmento correspondiente a los procesos registrados solo una vez representa el 10.4 % del total de MIPS consumidos hasta la fecha. Por su parte, los procesos registrados en todas las fechas han consumido el 38.8 % del total. Finalmente, los procesos con un número intermedio de registros han contribuido con el 50.8 % del consumo total de MIPS. En la siguiente sección, se presentarán las cantidades totales de estas ejecuciones y consumos para proporcionar un contexto numérico más preciso.

4.2.2. Análisis de las Variables Cuantitativas

Al analizar las variables cuantitativas (`total_ejecucionesFecha` y `total_mipsFecha`) sin agrupar por ninguna variable cualitativa ni por la fecha, los resultados del recuento, la suma total, las medidas de tendencia central y dispersión, presentados en la Tabla 11, evidencian que, en general, los procesos se ejecutan pocas veces y el consumo de MIPS es bajo. En particular, el análisis muestra que, en promedio, cada proceso utiliza menos de 1 MIPS al día, lo que sugiere una distribución altamente sesgada, con la mayoría de los valores cercanos a cero. Además, aunque la media de las ejecuciones diarias por proceso es de 298, la mediana es de solo 2, lo que indica que la mayoría de los procesos tienen un número reducido de ejecuciones, mientras que un pequeño grupo de procesos presenta valores excepcionalmente altos.

Estadística	Variable	
	`total_ejecucionesFecha`	`total_mipsFecha`
Recuento	2,530,760	2,530,760
Suma	751,537,512	105,866
Media	298	0.04
Mediana	2	0.00003
Mínimo	1	0
Máximo	147,842	82.37
Des. Estándar	3403	0.46

Tabla 11: Medidas de Tendencia Central y Disperción de las Variables Cuantitativas.

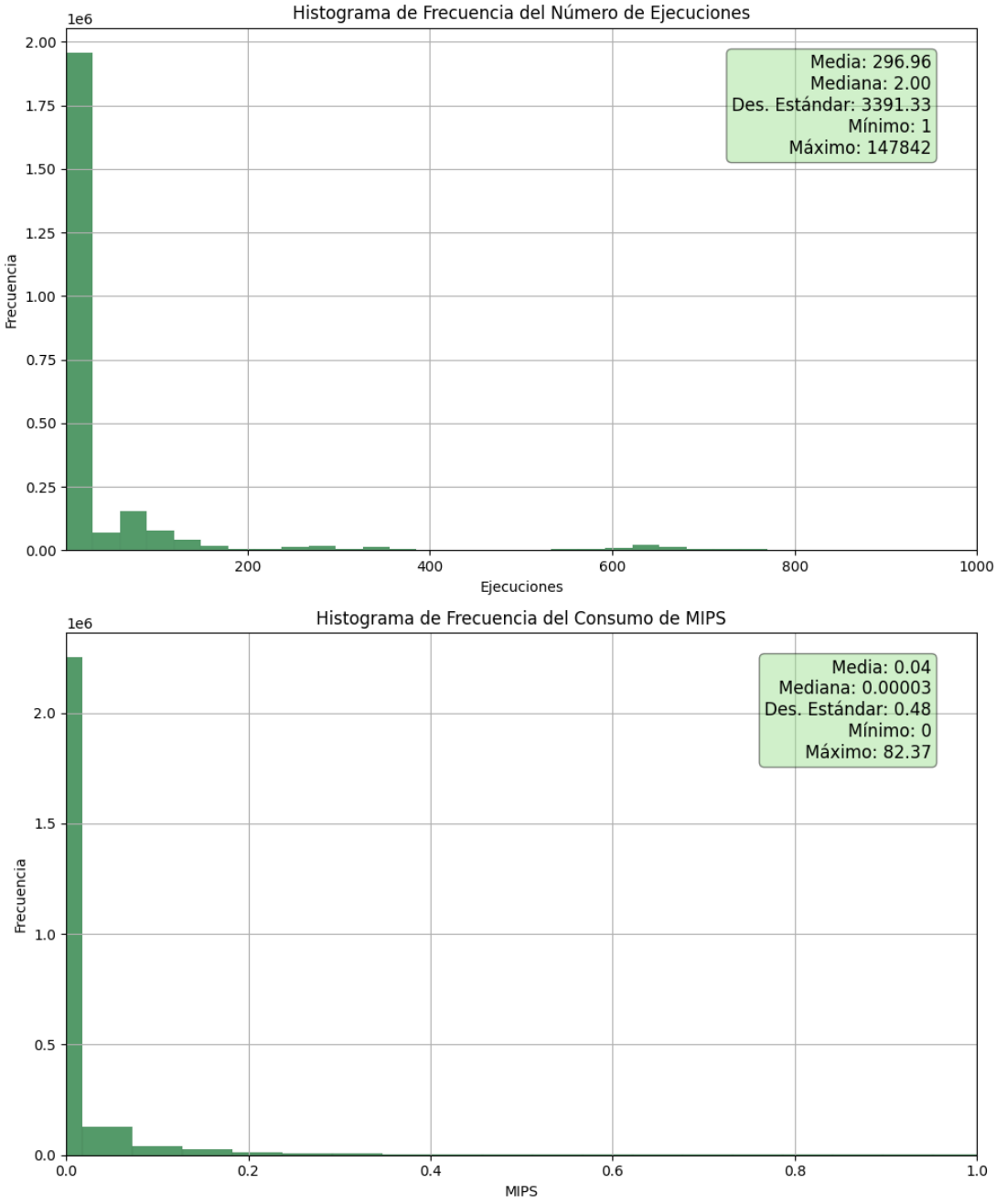

Figura 13: Histogramas de las Variables Cuantitativas.

La Figura 13 contiene dos histogramas que representan la distribución de `total_ejecucionesFecha` y `total_mipsFecha`, respectivamente. En ambos casos, los valores máximos han sido truncados en la visualización, ya que las observaciones con valores extremadamente altos son muy poco frecuentes y su

presencia no afecta significativamente la distribución general.

El primer histograma muestra la distribución del número de ejecuciones por proceso. Se observa una distribución altamente sesgada a la derecha, con la mayoría de los valores concentrados en niveles bajos. La media es de aproximadamente 298.06 ejecuciones por proceso, mientras que la mediana es significativamente menor (2.00), lo que indica que la mayoría de los procesos tienen pocas ejecuciones, pero existen algunos con valores extremadamente altos que elevan la media. Además, la desviación estándar es 3402.52, lo que confirma una gran dispersión en los datos. Aunque el máximo registrado es de 147,842 ejecuciones, estos casos son atípicos y ocurren con una frecuencia cercana a cero. La gran concentración de datos en valores bajos y la presencia de valores extremos sugieren una distribución altamente desigual en la cantidad de ejecuciones entre procesos.

El segundo histograma representa la distribución del consumo de MIPS por proceso y exhibe un comportamiento similar al del número de ejecuciones. La distribución está fuertemente sesgada a la derecha, con la mayoría de los valores cercanos a cero. La media es 0.04 MIPS, mientras que la mediana es extremadamente baja (0.00003 MIPS), lo que indica que en la mayoría de los casos los procesos consumen cantidades insignificantes de MIPS. La desviación estándar de 0.46 MIPS sugiere la presencia de procesos con un consumo relativamente superior al promedio. El máximo consumo registrado es 82.37 MIPS, pero, al igual que en el caso anterior, estos valores son eventos aislados con una frecuencia casi nula. La forma de la distribución indica que solo unos pocos procesos tienen un consumo elevado, mientras que la gran mayoría presenta valores cercanos a cero.

En términos generales, ambas variables presentan distribuciones con una marcada asimetría, donde la mayoría de los procesos tienen valores bajos tanto en número de ejecuciones como en consumo de MIPS, pero existen casos excepcionales con valores extremadamente altos. **Esto sugiere que un pequeño subconjunto de procesos es responsable de un uso desproporcionado de los recursos**, lo que puede ser un factor clave a considerar para la optimización del consumo de MIPS.

Siguiendo con el análisis, se decidió agrupar las variables cuantitativas por la variable `Fecha` y calcular la suma de sus valores para obtener tres nuevas métricas que representan el consolidado diario: la suma total de ejecuciones, denominada `Total_Ejecuciones`; la suma total del consumo de MIPS, denominada `Total_MIPS`; y el conteo total de registros, denominado `Nro_Registros`, que indica el número total de procesos únicos activados en cada día. Estos valores permiten evaluar la carga operativa diaria y la variabilidad en el consumo de recursos.

Estadística	Variable		
	`Total_Ejecuciones`	`Total_MIPS`	`Nro_Registros`
Recuento	956	956	956
Media	786,127.10	110.74	2,647.24
Mediana	810,642.50	106.76	2,686
Mínimo	348,057	43.76	1,758
Máximo	1,670,946.00	253.38	3,066
Des. Estándar	70,136.74	24.65	161.54

Tabla 12: Medidas de Tendencia Central y Dispersión del Consolidado Diario.

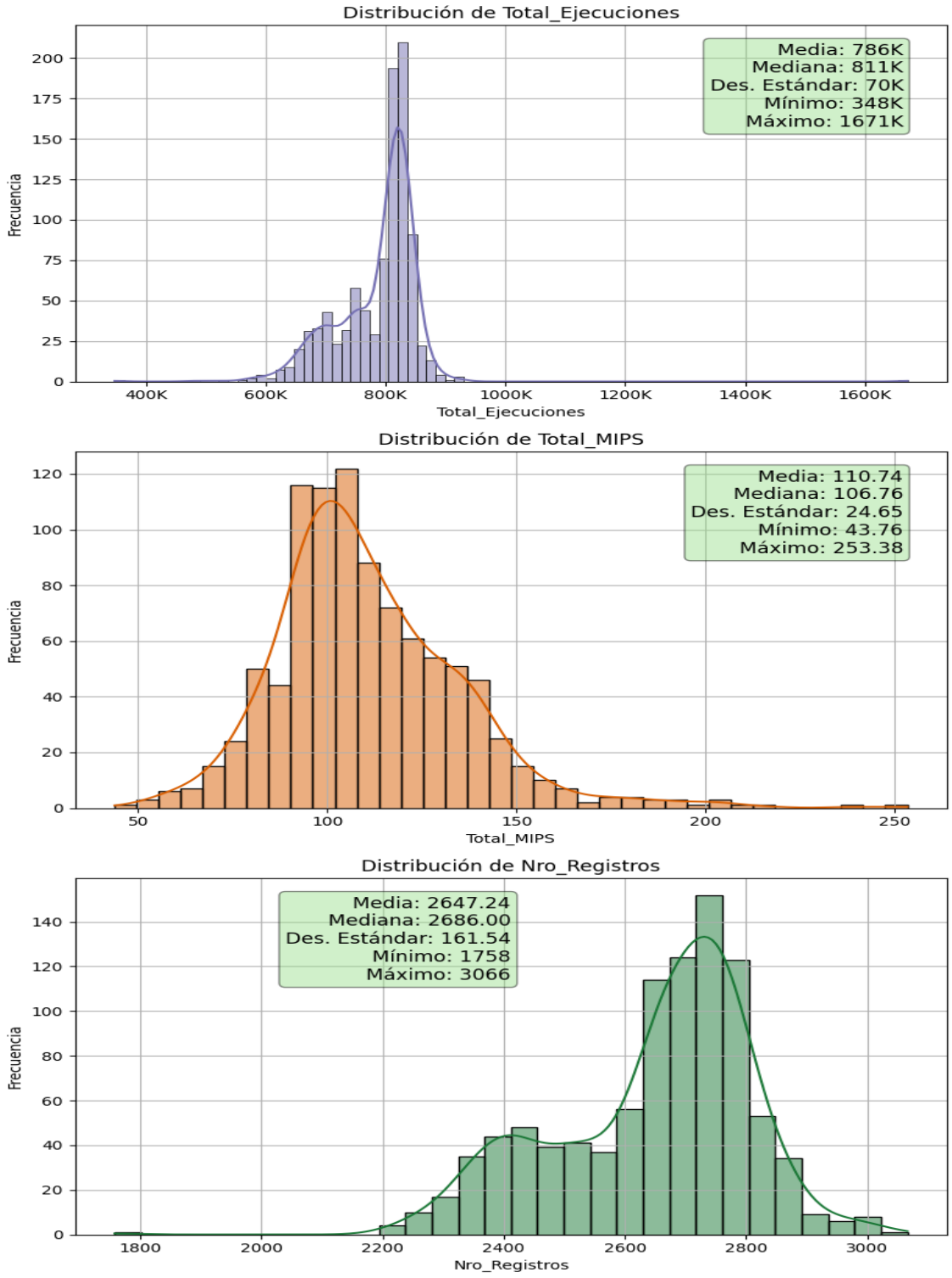

Figura 14: Histogramas de las Variables Cuantitativas Agrupadas Por `Fecha`.

En la Tabla 12, se presentan las principales medidas descriptivas de estas variables. Se observa que el número total de ejecuciones diarias (`Total_Ejecuciones`) tiene un valor medio de aproximadamente 786,127 ejecuciones, con una mediana de 810,642.50, lo que indica una distribución con ligera asimetría. El mínimo registrado es de 348,057 ejecuciones, mientras que el máximo alcanza 1,670,946, con una desviación estándar de 70,136.74, lo que sugiere una variabilidad moderada.

El consumo total de MIPS por día (`Total_MIPS`) presenta una media de 110.74 y una mediana de 106.76, lo que sugiere una distribución relativamente simétrica. Los valores extremos van desde 43.76 hasta 253.38, con una desviación estándar de 24.65, lo que indica que la mayoría de los valores se agrupan en torno a la media, pero con algunos días de mayor consumo.

Por otro lado, la cantidad total de procesos únicos activados por día (`Nro_Registros`) tiene una media de 2,647.24, con una mediana de 2,686. El rango de valores oscila entre 1,758 y 3,066, con una desviación estándar de 161.54, lo que muestra una distribución relativamente estable con variaciones moderadas.

Las distribuciones de estas tres métricas se representan en los histogramas de la Figura 14, donde se pueden observar los patrones de concentración y dispersión. En el caso de `Total_Ejecuciones`, la distribución muestra una alta concentración en torno a la media, con algunos días atípicos de mayor número de ejecuciones. La variable `Total_MIPS` sigue un patrón similar, con una forma cercana a la normalidad, mientras que `Nro_Registros` exhibe una ligera asimetría hacia valores más altos.

Estos resultados proporcionan una visión detallada del comportamiento diario de la plataforma, permitiendo identificar patrones y posibles anomalías en la ejecución de procesos y el consumo de MIPS.

De la Figura 14 se observa que la distribución del número de registros diarios es asimétrica hacia la izquierda (sesgo negativo), con un pico predominante entre 2600 y 2800 registros. En general, los datos muestran cierta variabilidad, pero sin cambios abruptos. Sin embargo, la presencia de valores más bajos puede sugerir posibles fallos en la recolección de datos, días festivos o menor actividad operativa en ciertos períodos.

Dado que Grupo Éxito opera acorde al calendario colombiano, es razonable suponer que los días festivos y fines de semana pueden influir en el gasto de recursos de operación, afectando el número de registros diarios. La discusión previa sobre la variabilidad en los registros, junto con la posible relación con la actividad operativa, motivó un análisis más detallado. Para profundizar en este aspecto, se exploró la distribución del número de registros segmentados por el día de la semana, con el objetivo de identificar patrones recurrentes y evaluar cómo varía la demanda de recursos en función del calendario laboral. Los resultados se pueden observar en la Figura 15.

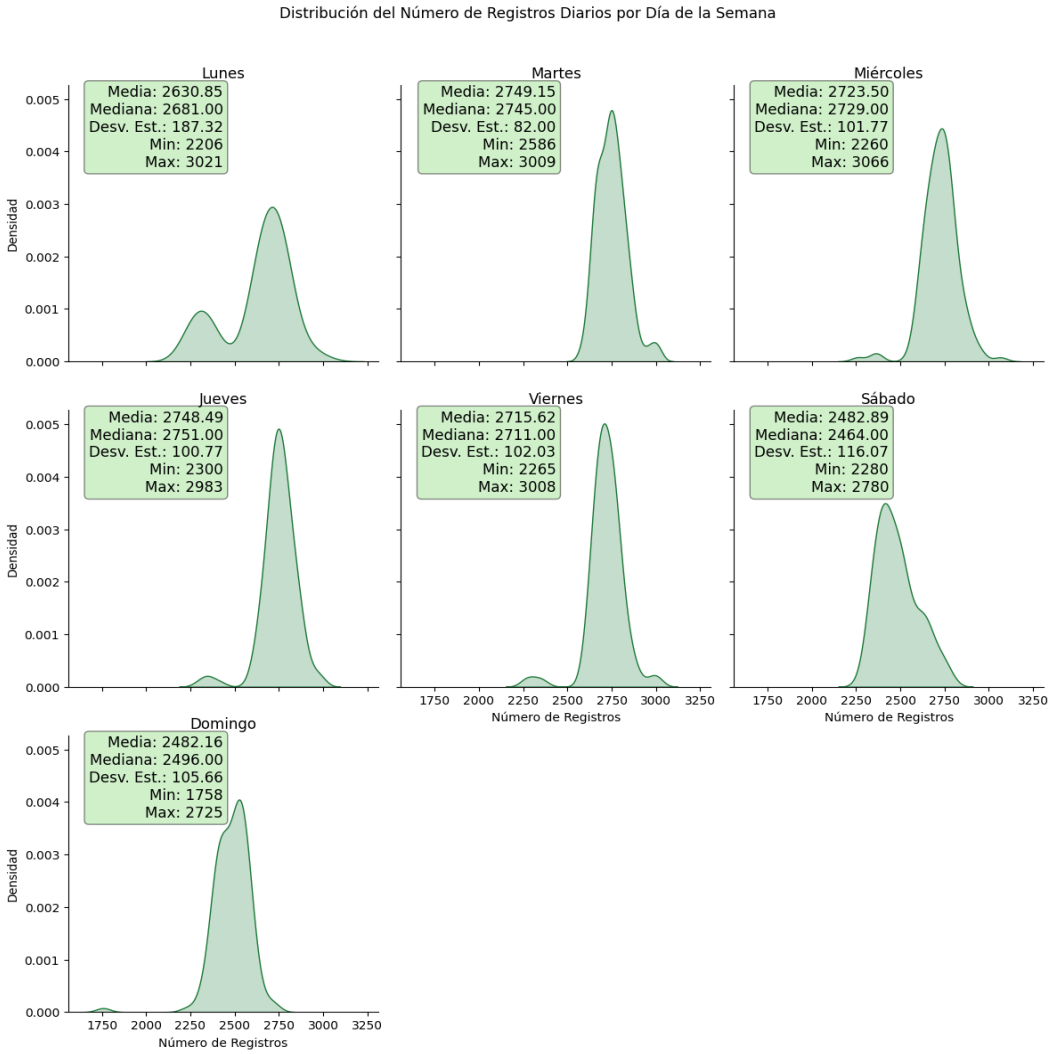

Figura 15: Densidad estimada de la distribución del número de registros segmentada por los días de la semana.

El análisis de la distribución del número de registros diarios por día de la semana (Ver Figura 15) indica que los días hábiles (martes a viernes) presentan una mayor concentración de registros, con valores promedio superiores a 2700, mientras que los fines de semana (sábado y domingo) exhiben una reducción con promedios en el rango de 2460-2490. Los lunes muestran una variabilidad más alta en comparación con el resto de los días, lo que podría atribuirse tanto a un posible efecto de acumulación de operaciones tras el fin de semana como al hecho de que, en el calendario colombiano, la mayoría de los días festivos ocurren el lunes, afectando así la actividad operativa. La disminución en el número de registros en ciertos días sugiere un patrón asociado a la menor demanda de servicios, la reducción de operaciones o la incidencia de días festivos. Estos hallazgos respaldan la sospecha de que la actividad operativa de Grupo Éxito sigue el calendario laboral, con un uso más intensivo de los recursos en días hábiles y una disminución notable durante los fines de semana y festivos.

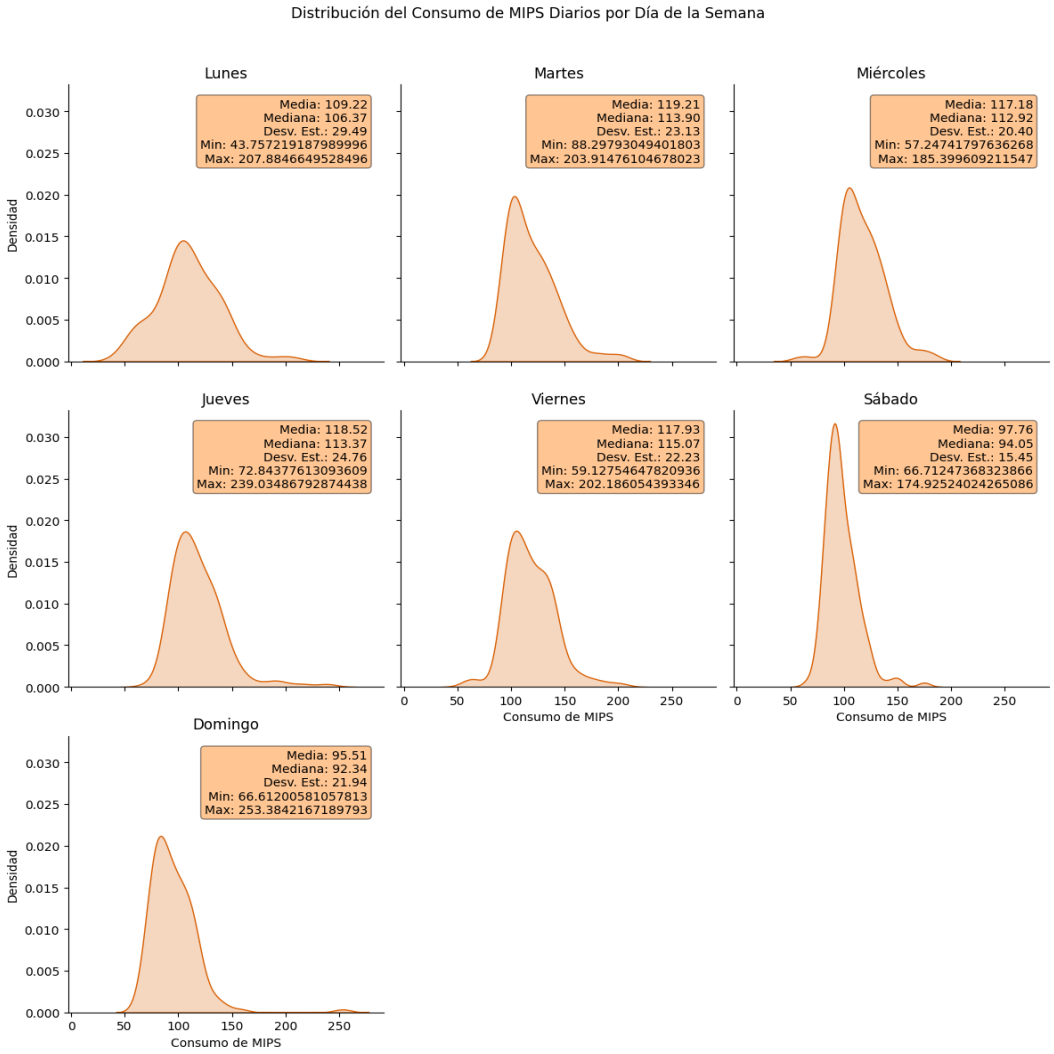

Figura 16: Densidad estimada de la distribución del consumo de MIPS segmentada por los días de la semana.

El análisis de la distribución del consumo diario de MIPS por día de la semana (Ver Figura 16) muestra que los días hábiles (lunes a viernes) presentan un consumo promedio superior a 109 MIPS, con una menor dispersión en comparación con los fines de semana. En particular, los martes y jueves exhiben los valores medios más altos, con 119.21 y 118.52 MIPS, respectivamente. Durante los fines de semana, el consumo disminuye significativamente, con valores promedio de 97.76 MIPS el sábado y 95.51 MIPS el domingo, reflejando una menor carga operativa en estos días. Se observa una mayor variabilidad en el consumo de MIPS los lunes, lo que podría estar relacionado con un efecto de acumulación tras el fin de semana o con la incidencia de días festivos. Este patrón sugiere que el uso de los recursos tecnológicos sigue el ciclo laboral de la compañía, con una mayor demanda en días hábiles y una reducción durante los fines de semana.

El análisis de correlación de las variables, representado mediante un diagrama de calor (Ver Figura 17),

muestra las relaciones entre las variables `Total_Ejecuciones`, `Total_MIPS` y `Nro_Registros`, donde se observa una correlación moderada entre `Total_Ejecuciones` y `Total_MIPS` (0.4), una relación más fuerte entre *Total_Ejecuciones* y `Nro_Registros` (0.62), y la correlación más alta entre `Total_MIPS` y `Nro_Registros` (0.67), lo que sugiere que el consumo de MIPS está estrechamente relacionado con el número de registros procesados. El diagrama de dispersión (Ver Figura 18) refuerza estos hallazgos y muestra que *Total_Ejecuciones* y `Total_MIPS` tienen una relación positiva pero con dispersión considerable, mientras que `Total_MIPS` y `Nro_Registros` presentan una mayor alineación lineal, lo que refuerza su alta correlación. En conclusión, `Nro_Registros` se perfila como el mejor predictor del consumo de MIPS.

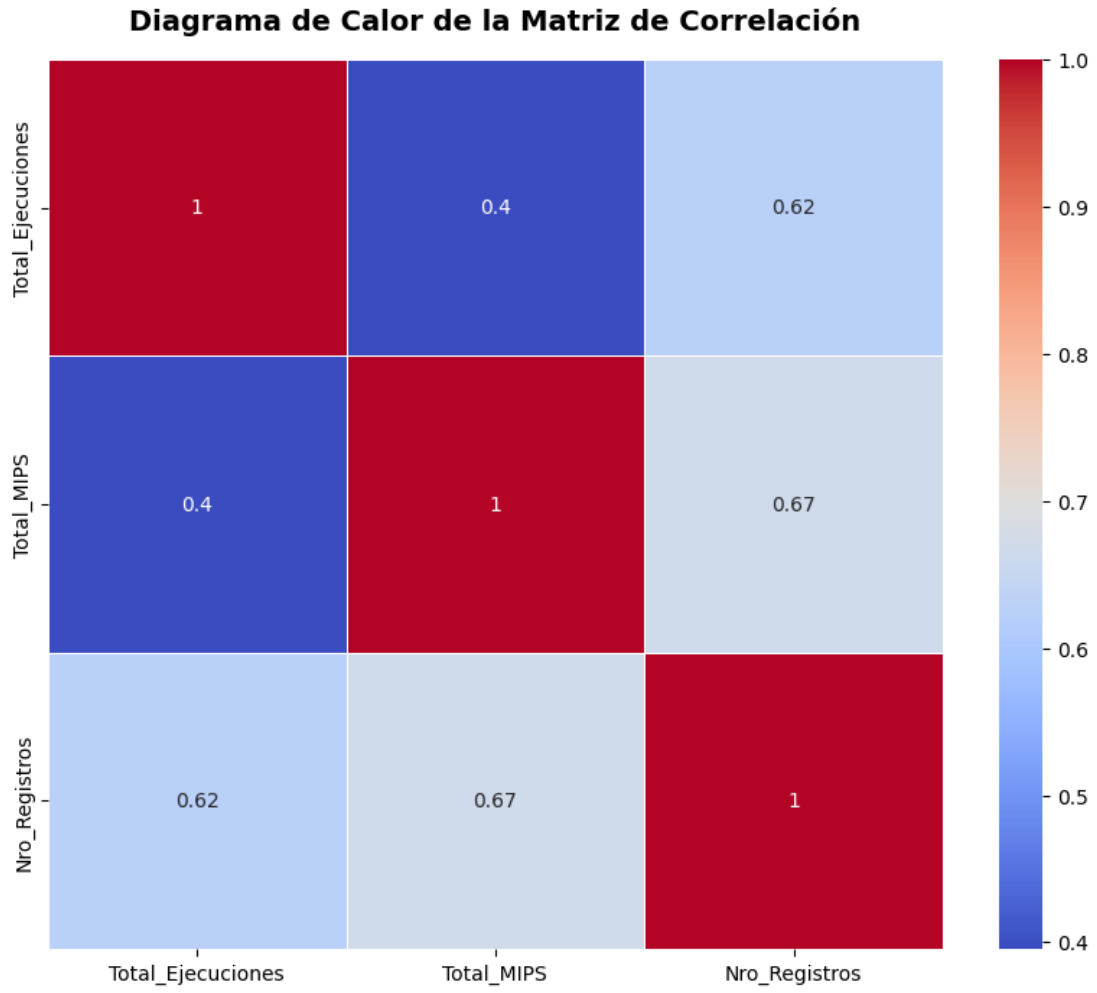

Figura 17: Diagrama de Calor de la Matriz de Correlación.

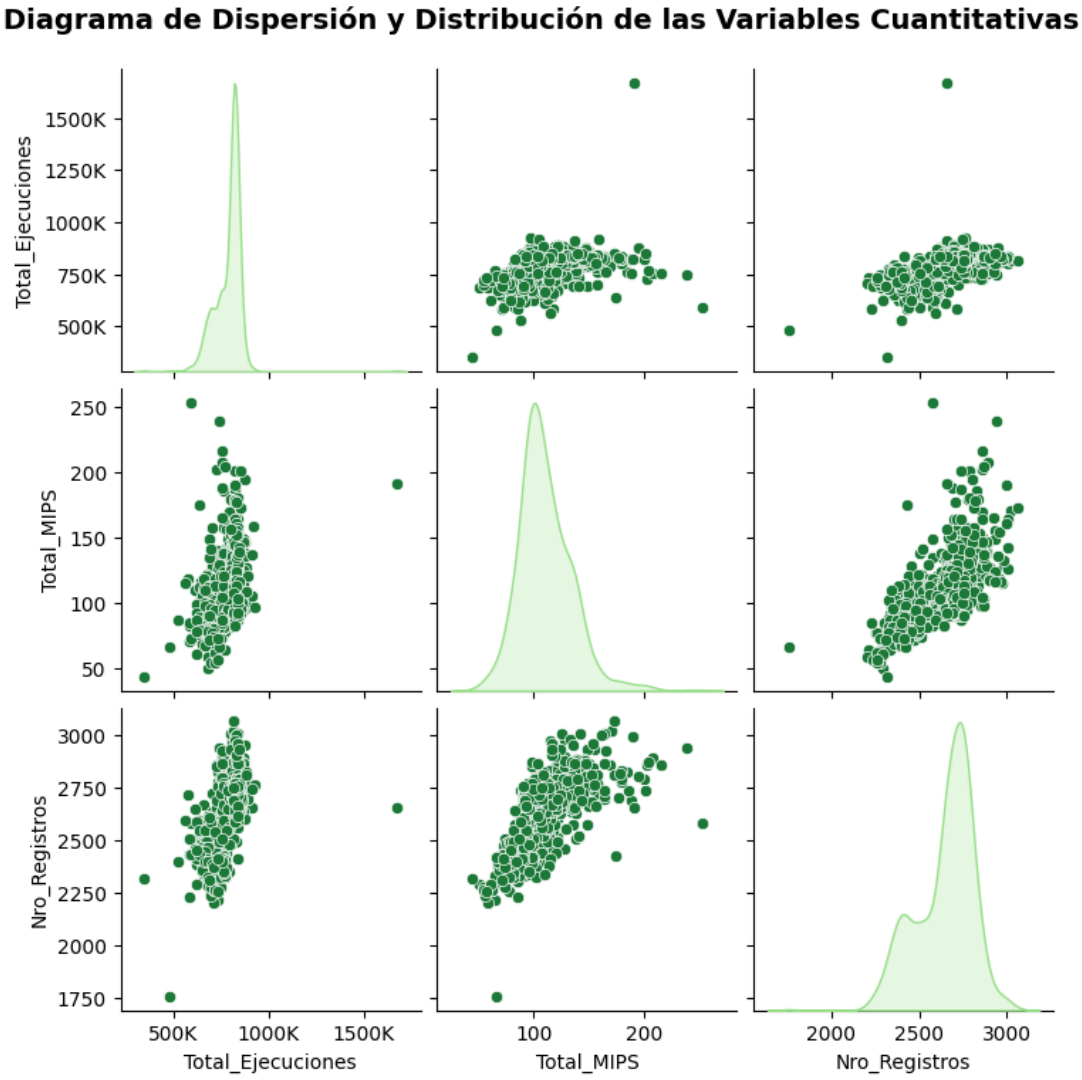

Figura 18: Diagramas de Dispersión Entre las Variables Cuantitativas.

La Figura 19 muestra una serie de tiempo multivariada conteniendo las variables `Total_Ejecuciones`, `Total_MIPS` y `Nro_Registros`, respectivamente. Se observa una periodicidad clara en las tres series, con fluctuaciones diarias y picos en momentos específicos. En la primera serie, `Total_Ejecuciones`, se aprecia una variación estable en la mayoría del período analizado, con un incremento brusco y una caída atípica alrededor de finales de 2023 e inicios de 2024. En la segunda serie, `Total_MIPS`, se pueden identificar eventos de alta variabilidad con valores extremos en ciertos días. Finalmente, la tercera serie, `Nro_Registros`, presenta un patrón más regular con algunas caídas abruptas en momentos específicos. Estos comportamientos permiten identificar posibles anomalías en la ejecución de procesos y en el consumo de recursos de la plataforma (Ver Sección 2.3).

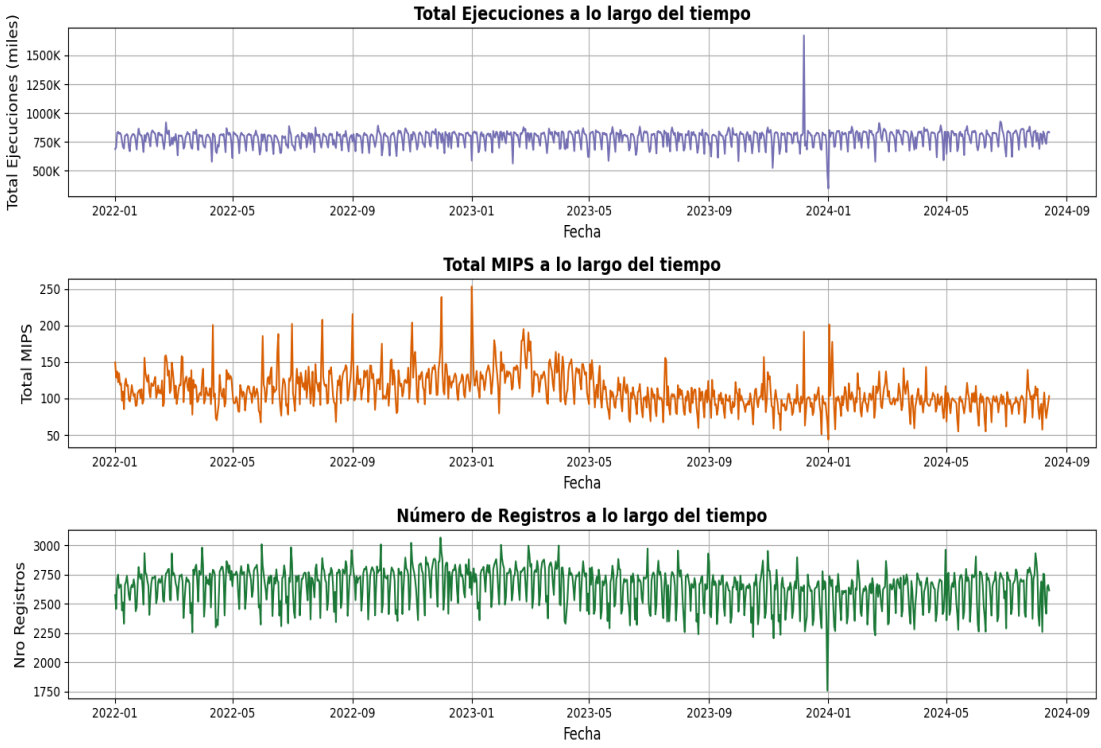

Figura 19: Serie de Tiempo Multivariada.

El análisis exploratorio de datos realizado sobre la tabla `dbo.refrescarprocesos_10dias` de la base de datos `WLMDW_TEMPO` permitió identificar patrones y comportamientos clave en el consumo de MIPS y la ejecución de procesos. Los principales hallazgos son los siguientes:

Estructura de la Tabla: La tabla contiene información relevante sobre el consumo de MIPS y el número total de ejecuciones, segmentados por nombre del proceso, nombre del grupo y fecha. Las restricciones de la tabla aseguran la exactitud y la completitud de los datos.

Distribución de Registros: La mayoría de los procesos tienen pocas activaciones, mientras que un subconjunto reducido se registra de manera constante en todas las fechas. Esto sugiere una estructura operativa dividida entre procesos críticos y eventuales.

Consumo de Recursos: Un pequeño subconjunto de procesos es responsable de un uso desproporcionado de los recursos, lo que puede ser un factor clave para la optimización del consumo de MIPS.

Patrones Diarios: Los días hábiles presentan una mayor concentración de registros y consumo de MIPS, mientras que los fines de semana y días festivos muestran una reducción significativa. Esto refleja el ciclo laboral de la compañía.

Correlación de Variables: Existe una correlación moderada entre el número de ejecuciones y el consumo de MIPS, y una relación más fuerte entre el número de registros y el consumo de MIPS, sugiriendo que el número de registros es una variable relevante para predecir el consumo de MIPS.

Series Temporales: La serie de tiempo multivariada muestra una periodicidad clara con fluctuaciones diarias y picos en momentos específicos, permitiendo identificar posibles anomalías en la ejecución de

procesos, el consumo de recursos o el número de registros ingresados.

En resumen, el análisis proporciona una visión detallada del comportamiento diario de la plataforma, identificando patrones operativos y posibles áreas de optimización en el consumo de MIPS.

4.3. Normalización de la Tabla `refrescarprocesos_10dias`

La normalización de la tabla `refrescarprocesos_10dias` se llevó a cabo con el objetivo de reducir la redundancia de datos y mejorar la integridad de la información dentro del esquema de base de datos. La estructura original de la tabla almacenaba información sobre la ejecución de procesos, registrando el número total de ejecuciones y el consumo de MIPS por fecha, junto con los nombres de los procesos y los grupos a los que pertenecen. Sin embargo, la presencia de datos categóricos como `NombreProceso` y `NombreGrupo` en una única tabla generaba redundancia y dificultaba su escalabilidad. La normalización permitió descomponer esta información en entidades separadas, minimizando la duplicación de datos y mejorando la eficiencia en consultas y análisis históricos.

La Figura 20 muestra la estructura resultante tras la normalización. En esta nueva organización, la información se distribuyó en entidades separadas con claves primarias y relaciones bien definidas:

- **Consumo de MIPS**: Centraliza la información sobre el consumo de recursos computacionales, incluyendo identificadores de proceso, grupo, fecha, día de la semana y categoría de atipicidad.
- **Predicciones MIPS**: Contiene las predicciones de consumo junto con los intervalos de confianza estimados.
- **Métricas de Predicción**: Almacena los valores de error asociados a los modelos de predicción, reduciendo la duplicación de métricas en cada registro.
- **Procesos y Grupos**: Se crearon tablas separadas para almacenar los nombres de los procesos y grupos, estableciendo una relación muchos a muchos mediante la tabla intermedia `ProcesosGrupos`.
- **Dimensiones de Fecha y Día de la Semana**: Se introdujeron tablas para normalizar la información temporal, lo que permite realizar consultas eficientes y facilita el análisis de tendencias.

Este proceso de normalización permitió mejorar la eficiencia en la consulta y actualización de los datos, garantizando una mayor consistencia en la información almacenada. La estructura final sigue los principios de normalización, logrando una base de datos más estructurada y optimizada para análisis y predicciones de consumo de MIPS.

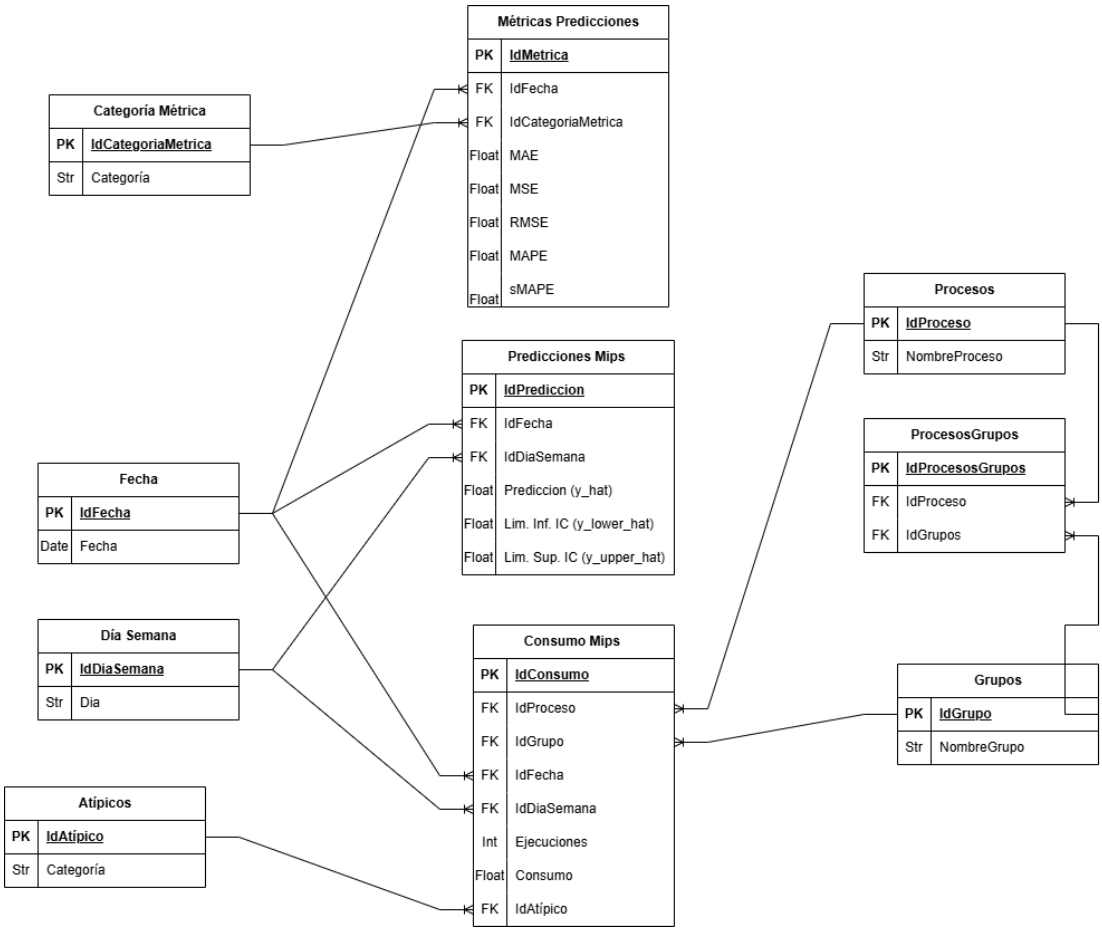

Figura 20: Esquema normalizado de la base de datos.

4.4. Arquitectura del Flujo de Trabajo del Microservicio

El diseño de la arquitectura del flujo de trabajo del microservicio (Ver Figura 21) se basa en la extracción, procesamiento y almacenamiento de datos relacionados con el consumo de MIPS. La fuente principal de datos es la base de datos de origen `WLMDW_TEMPO`, que contiene la tabla `dbo.refrescarprocesos_10dias`.

La extracción de datos se realiza mediante una conexión ODBC establecida desde un contenedor de Docker, el cual ejecuta una aplicación desarrollada en lenguaje Python. Esta aplicación se ejecuta diariamente a las 9:30 AM a través de un `CronJob` alojado en el repositorio `//pdn-mips-sinco-estadísticas`. El contenedor proporciona un entorno controlado para garantizar la ejecución eficiente del procesamiento de datos.

Dentro del proceso de transformación, la aplicación Python realiza varias tareas fundamentales: la normalización de los datos para evitar redundancias, la detección de anomalías en el consumo de MIPS y la estimación de valores futuros mediante modelos de predicción. Estos datos procesados se almacenan en la base de datos de destino `Consumos-PrediccionesMIPS`, alojada en el servidor `296SQLP06`. Esta base de datos contiene múltiples tablas interrelacionadas que permiten organizar la información de manera

estructurada y facilitar su consulta posterior desde el tablero de *Power BI*.

Finalmente, los datos almacenados en la base de datos de destino son utilizados en Power BI para la generación de tableros interactivos. Estos tableros permiten al personal de operaciones e infraestructura monitorear el consumo de MIPS, identificar posibles anomalías y tomar decisiones informadas.

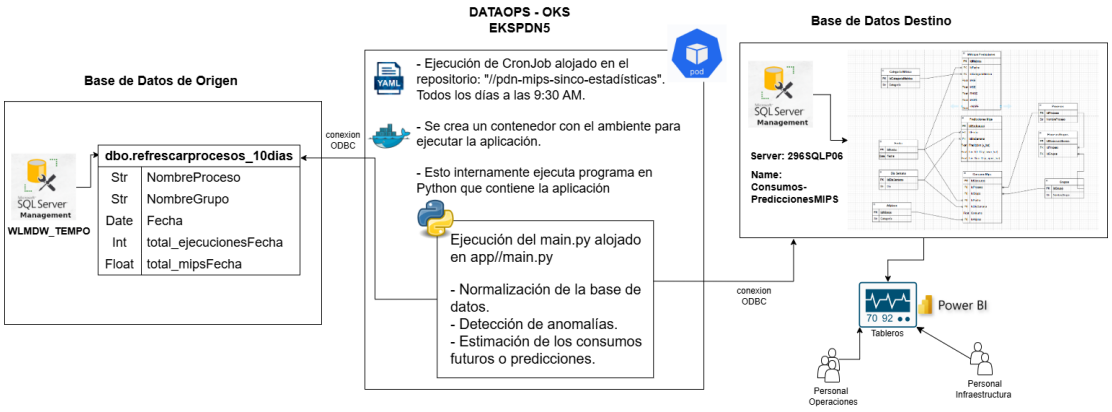

Figura 21: Arquitectura del Flujo de Trabajo del Microservicio.

4.5. Desarrollo de Modelos y Construcción de la Base de Datos

Para garantizar un análisis eficiente del consumo de MIPS, se desarrollaron e integraron múltiples componentes dentro del flujo de trabajo del microservicio. Inicialmente, se implementó un modelo de detección de datos atípicos. Posteriormente, se construyó un modelo predictivo basado en *PROPHET*, con el objetivo de estimar el consumo futuro de MIPS con base en patrones históricos. Paralelamente, se diseñó y creó la base de datos `Consumo-PrediccionesMIPS`, siguiendo una estructura normalizada que facilita el almacenamiento y la consulta eficiente de los resultados. Finalmente, se desarrolló un módulo de integración que automatiza la extracción, procesamiento e inserción de datos en las tablas correspondientes, consolidando así un sistema robusto para la gestión y análisis del consumo de MIPS.

```
`/mips-sinco-estadísticas/
    ├── app/
    |   ├── database_tools/
    |   |       ├── init.py
    |   |       ├── connections.py
    |   |       ├── create_tables.py
    |   |       ├── delete_tables.py
    |   |       └── update_tables.py
    |   ├── forecast_tools/
    |   |       ├── init.py
    |   |       └── metrics.py
    |   ├── main_functions/
    |   |       ├── init.py
    |   |       ├── forecasting.py
    |   |       ├── inserting_data.py
    |   |       └── novelty_detection.py
    |   └── main.py
```

Figura 22: Esquema modular que representa la organización de los archivos de *Python* dentro del directorio `app` que contiene el *software* de extracción, transformación y carga de datos.

La construcción del modelo detector de anomalías comenzó por definir la variable `total_mipsFecha` como la variable de interés para el análisis. Por lo anterior, el tipo de dato de entrada del modelo de detección de anomalías de este trabajo es una serie de tiempo univariada. El tipo de dato atípico en el que se estaba interesado era de tipo puntual. El orden temporal de los datos no se tuvo en cuenta en el análisis. El modelo se diseñó para aplicarse en flujo no continuo (*Non-Streaming*) y continuo (*Streaming*). La primera característica se aplicó la primera vez que el modelo se ejecutó, esto con el propósito de categorizar los consumos de MIPS que ya estaban almacenados en la base de datos, por lo que el modelo se denomina **modelo de estimación** en esta primera etapa; la segunda característica se aplicó luego de que la primera etapa estuvo finalizada, esto se debía a que en esta segunda etapa se debían categorizar los nuevos consumos que llegaban a la base de datos en cuanto llegaban, por lo anterior, el modelo se denomina **modelo de predicción**.

En ambas etapas se aplicó el método del Rango Intercuartílico (IQR) (Ver Sección 2.4) para los procesos con una cantidad de registros superior a 27 y que mostraban provenir de una distribución normal después de aplicar la prueba de Shapiro-Wilk con un nivel de significancia $\alpha = 0.05$. Por otro lado, se aplicó el método de la Desviación Absoluta Mediana (Ver Sección 2.5) para los procesos con una cantidad de registros en el rango 6-27 o los procesos con una cantidad de registros superior a 27 pero que no mostraban provenir de una distribución normal. Los procesos con una cantidad de registros en el rango de 1-5 se agruparon según la cantidad de registros, creándose así 5 grupos donde cada grupo estaba relacionado por la cantidad de registros que tenían, es decir, los procesos que tenían un solo registro formaban un grupo, los procesos que tenían dos registros formaban un segundo grupo y así sucesivamente. A cada uno de estos grupos se les aplicó el método de la Desviación Absoluta Mediana.

Dado que no se quería categorizar el consumo de MIPS de un registro como atípico a no ser que estuviera muy por encima de su valor esperado, se decidió utilizar $k = 3$ para ambas metodologías: IQR y MAD. El archivo `novelty_detection.py` del módulo `main_functions/` (Ver Figura 22) contiene el código en lenguaje *Python* para ejecutar todos los pasos anteriormente mencionados.

El modelo de predicción de consumo de MIPS se implementó en el archivo `forecasting.py` del módulo `main_functions/` utilizando la biblioteca `PROPHET`. Dado que el objetivo era proyectar el consumo total de MIPS diario en lugar de analizar procesos individuales, se agruparon los datos por la variable `Fecha` y luego se sumaron los consumos de MIPS, obteniendo así el consolidado diario. Después de tener preparados los datos en el formato admitido por el modelo `PROPHET`, se ajustó el modelo teniendo en cuenta la Ecuación 2 y los días festivos del calendario colombiano. Para evaluar el desempeño del modelo, se creó el archivo `metrics.py`, el cual calcula las métricas de error (Ver Sección 2.7) establecidas en el Objetivo del Microservicio (Ver Sección 3.2), proporcionando así una referencia cuantitativa sobre la calidad de las estimaciones generadas.

La gestión de la base de datos destino `Consumo-PrediccionesMIPS` se realizó a través de los archivos `create_tables.py` y `delete_tables.py` del módulo `database_tools/`. En `create_tables.py`, se definieron las estructuras normalizadas de las tablas siguiendo el esquema diseñado (Ver Figura 20), garantizando una organización eficiente de los datos. Durante la fase de pruebas, el archivo llamado `delete_tables.py` permitió la eliminación y reconstrucción de las tablas para validar la correcta implementación de los modelos y la integridad de la información almacenada. Todas las conexiones a la base de datos fueron manejadas a través del archivo `connections.py` del módulo `database_tools/`, centralizando la gestión de credenciales y parámetros de conexión para garantizar la seguridad y estabilidad en las interacciones con el servidor de bases de datos.

El flujo completo de manipulación de datos se consolidó en el archivo `inserting_data.py` del módulo `main_functions/`, el cual se encarga de extraer los registros relevantes desde la base de datos de origen, transformarlos según los modelos de predicción y detección de anomalías, e insertarlos en la base de datos de destino. Este módulo automatiza el proceso, reduciendo la intervención manual y permitiendo

una actualización continua de la información procesada.

Finalmente, el archivo `main.py` es el responsable de orquestar la ejecución correcta de todos los módulos, asegurando la integración y coordinación del flujo de trabajo. Gracias a esta arquitectura modular, se logró una solución escalable y eficiente para el análisis y la gestión del consumo de MIPS.

4.6. Desarrollo de Tablero de Monitoreo en Power BI

El tablero desarrollado tiene como objetivo principal proporcionar una visión integral del consumo de MIPS, permitiendo identificar patrones, tendencias y anomalías en el comportamiento de los procesos. Para ello, se han incorporado diversas visualizaciones y métricas que facilitan la interpretación de los datos.

En la sección **General**, que se puede observar en la Figura 23, se presenta un resumen global del consumo de MIPS, incluyendo las predicciones, la cantidad de recursos consumidos y habilitados, así como métricas de tendencia central. Además, se incorporan filtros por proceso y grupo, junto con una tabla que muestra los consumos mensuales históricos y las estimaciones para fechas futuras.

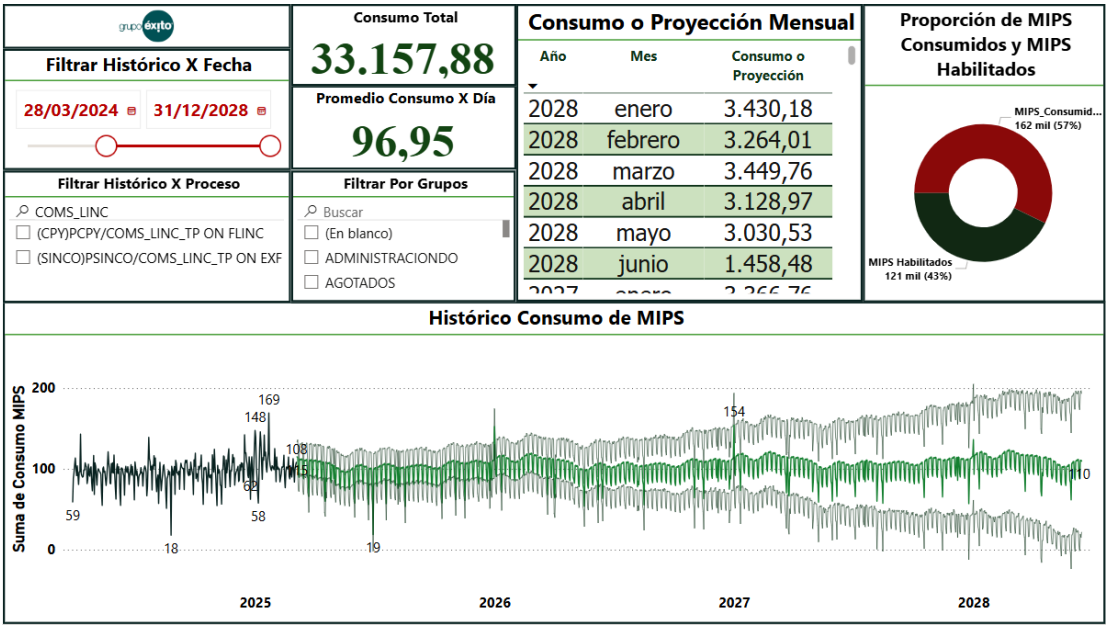

Figura 23: Ventana Principal del Tablero.

En la Figura 24 se muestra la sección de **Procesos Anómalos**, la cual permite identificar aquellos procesos cuyo consumo de MIPS ha sido clasificado como atípico. La información se presenta en una tabla donde los procesos se ordenan de mayor a menor consumo de MIPS, facilitando así la detección de los casos más críticos. Además, se dispone de un filtro por fecha para analizar períodos específicos y, al seleccionar un proceso, se genera un gráfico de líneas que muestra su comportamiento en el tiempo, permitiendo una mejor interpretación de su tendencia y posibles anomalías.

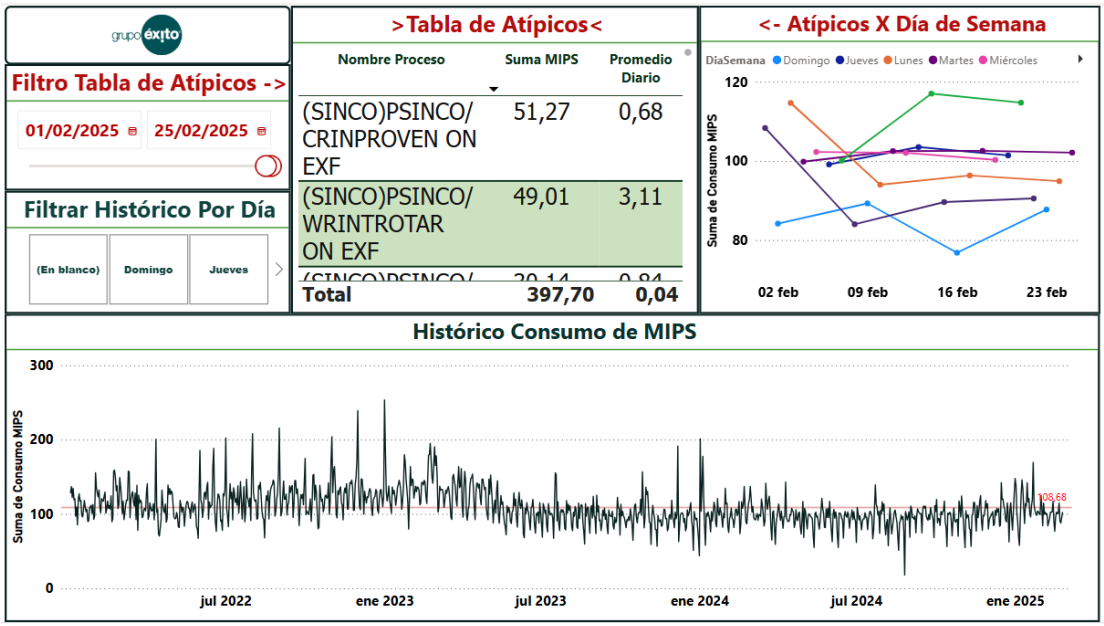

Figura 24: Ventana de los Procesos Anómalos Filtrados.

La Figura 25 muestra la sección de **Variables Explicativas** que permite analizar en profundidad las correlaciones entre las tres variables numéricas estudiadas en el análisis exploratorio de datos (Ver Sección 4.2). A diferencia de un análisis estático, esta sección proporciona herramientas interactivas que permiten filtrar los datos por proceso y grupo, facilitando la observación de cómo interactúan las variables en distintos contextos. Esta funcionalidad representa una ventaja significativa, ya que realizar este análisis de manera manual implicaría la construcción de código cada vez que se desee cambiar de proceso o grupo, lo que resultaría en un consumo de tiempo considerable. Adicionalmente, se incluyen tarjetas con los valores del coeficiente de correlación de Pearson para cada par de variables explicativas: el número de ejecuciones y el número de registros, en relación con la variable dependiente: el consumo de MIPS, lo que permite interpretar de manera más ágil la fuerza y dirección de estas relaciones.

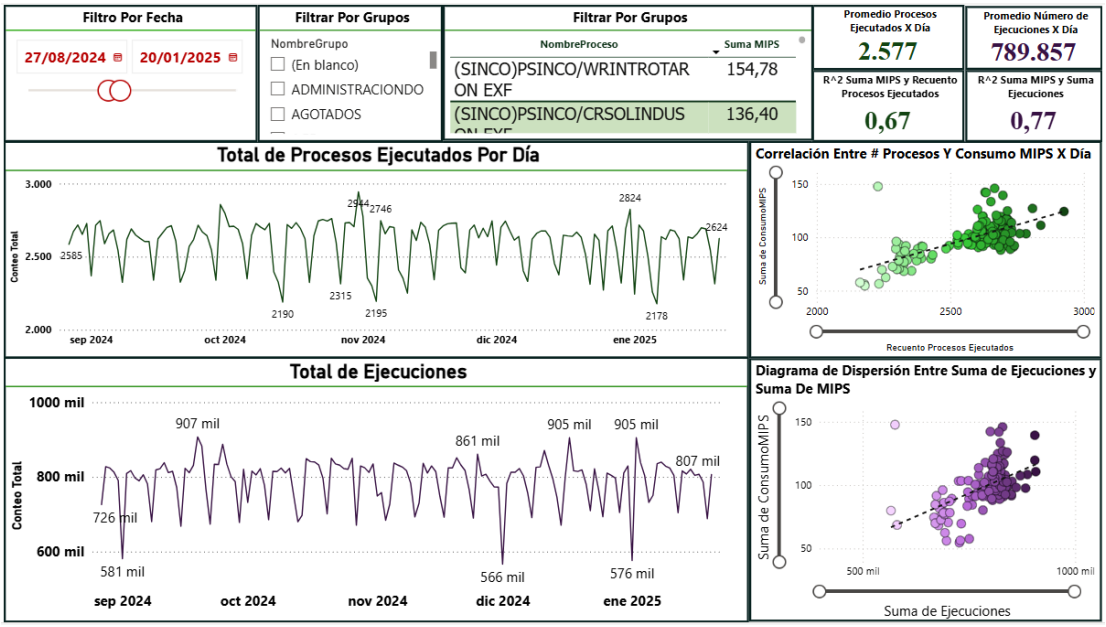

Figura 25: Ventana de las Correlaciones Entre el Consumo de MIPS y las variables Número de Registros y Número de Ejecuciones.

Finalmente, la sección de **Métricas de Precisión**, presentada en la Figura 26, ofrece una evaluación detallada del desempeño del modelo predictivo utilizado para estimar el consumo de MIPS. Las métricas han sido calculadas desde el 1 de noviembre de 2024 y se presentan en dos modalidades: mensual e histórica. La evaluación mensual permite analizar el comportamiento del modelo mes a mes, identificando posibles variaciones en su precisión, mientras que la evaluación histórica proporciona una visión acumulada del desempeño del modelo desde su fecha de inicio. Entre las métricas incluidas se encuentran el RMSE, MAE, MAPE y sMAPE, las cuales permiten cuantificar el nivel de error y la precisión de las predicciones en distintos horizontes temporales.

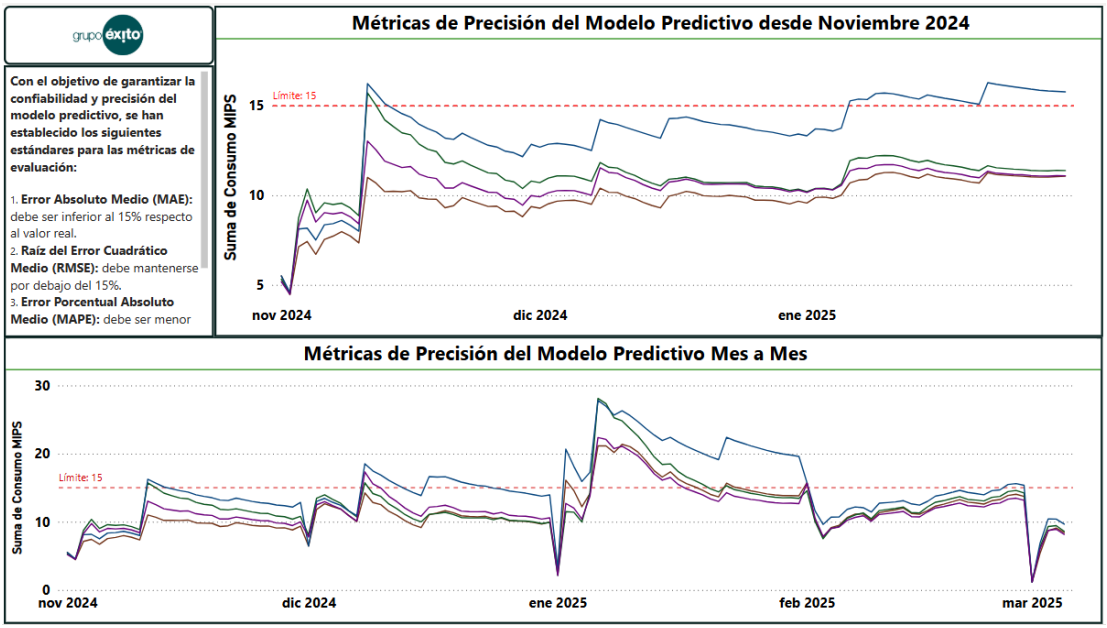

Figura 26: Ventana del la Métricas de Precisión del Modelo Predictivo.

Este tablero constituye una herramienta esencial para el monitoreo y la optimización del consumo de MIPS, permitiendo una gestión más eficiente de los recursos tecnológicos y contribuyendo a la mejora del rendimiento del sistema.

4.7. Resultados del Pipeline, CronJob y Dockerfile

Una vez construidos estos archivos siguiendo las pautas y plantillas de parte de la gerencia de TI de Grupo Éxito se obtuvo un repositorio cuyo directorio está ilustrado en la Figura 27.

Figura 27: Esquema General del Repositorio que Contiene el Microservicio.

A continuación, se da una explicación detallada de los archivos relevantes para esta sección:

La carpeta `pipeline` contiene los archivos necesarios para la configuración y ejecución del proceso de integración y despliegue continuo (CI/CD) en Azure DevOps. El archivo `vars-azure-pipeline.yml` define las variables utilizadas en el pipeline, estableciendo distintos grupos de configuración según la rama desde la que se ejecute: `develop` para desarrollo, `release` para calidad y `master` para producción. Además, se especifica la ruta de trabajo predeterminada para los archivos requeridos durante la ejecución. Por otro lado, el archivo `main-pipeline.yml` define la estructura del pipeline principal, extendiéndolo desde una plantilla preexistente ubicada en el repositorio `GCIT/arquitectura-documentacion`. Este archivo configura el entorno de ejecución para trabajar con *Python* en un entorno *DataOps* y despliega los componentes en `OKS` (OpenShift Kubernetes). Gracias a esta arquitectura modular, se garantiza una

integración automatizada y consistente en los diferentes entornos, minimizando la intervención manual y facilitando la administración del sistema.

La carpeta `charts` contiene los archivos necesarios para la configuración y despliegue del microservicio en un entorno Kubernetes mediante Helm. Dentro de esta carpeta, la subcarpeta `templates` alberga los archivos `cronjob.yaml` y `secrets.yaml`, encargados de la planificación de tareas y la gestión segura de credenciales, respectivamente.

El archivo `Chart.yaml` define la información general del paquete Helm, incluyendo el nombre del proyecto, su versión y una descripción de su propósito, que en este caso es la ejecución de procesos para el análisis de datos en MIPS. El archivo `values.yaml` almacena los valores predeterminados utilizados en la plantilla, definiendo variables clave como el entorno de despliegue, recursos asignados, programación del cronjob y parámetros de conexión a la base de datos.

El `cronjob.yaml` establece la ejecución periódica del microservicio mediante un `CronJob` de Kubernetes, el cual utiliza las variables definidas en `values.yaml` para determinar su configuración, como el entorno de ejecución, la imagen del contenedor, los recursos asignados y la política de reinicio. Este cronjob se encarga de extraer datos desde la base de datos de origen, procesarlos e insertarlos en la base de datos destino.

Para la gestión segura de credenciales, el archivo `secrets.yaml` define un objeto `Secret` en Kubernetes, que almacena las contraseñas necesarias para acceder a las bases de datos de extracción e inserción. Estos valores se inyectan dinámicamente en los contenedores del cronjob, garantizando la protección de la información sensible.

En conjunto, estos archivos permiten una implementación automatizada y segura del microservicio, asegurando su correcta ejecución en el entorno productivo sin intervención manual.

La configuración del entorno de ejecución del microservicio se realizó mediante contenedores Docker, utilizando los archivos `Dockerfile` y `docker-compose.yaml`. Estos archivos permiten la creación de una imagen de Docker con todas las dependencias necesarias y la ejecución del servicio en un entorno controlado y reproducible.

El archivo `Dockerfile` define la construcción de la imagen del contenedor, partiendo de una imagen base de `Ubuntu`. Se establece el directorio de trabajo en `/project` y se copian los archivos de la aplicación, incluyendo el directorio `app` y el archivo `requirements.txt`. Posteriormente, se instalan paquetes esenciales, como `curl`, `python3` y `pip`, asegurando que el entorno disponga de todas las herramientas necesarias para ejecutar el código. Se configura la zona horaria a `America/New_York` y se instalan certificados de seguridad para garantizar conexiones seguras. Además, se incluye la instalación del controlador ODBC de Microsoft (`msodbcsql17`), permitiendo la conexión con bases de datos SQL Server. Finalmente, el contenedor se configura para ejecutar el archivo `main.py` al iniciarse.

Por otro lado, el archivo `docker-compose.yaml` facilita la orquestación de los contenedores, definiendo un servicio llamado `test`, que se construye a partir del `Dockerfile` especificado. Se establece la versión `0.1.0` de la imagen y se define la variable de entorno utilizando un archivo `.env`, lo que permite una configuración más flexible sin necesidad de modificar el código fuente.

En conjunto, estos archivos permiten desplegar el microservicio de manera eficiente, garantizando un entorno homogéneo y portátil para su ejecución en distintos entornos sin depender de la infraestructura subyacente.

Como parte fundamental del desarrollo del microservicio, se elaboró una documentación completa que detalla su funcionamiento, implementación y uso. En la Figura 27 se observa la carpeta `documentation`, la cual contiene un archivo `readme.md` que proporciona una descripción estructurada del microservicio. Esta documentación incluye los requerimientos funcionales y no funcionales, justificando la necesidad

del microservicio dentro del sistema. Asimismo, se presenta la arquitectura del flujo de trabajo, permitiendo comprender cómo interactúan sus diferentes componentes. Además, se detalla la estructura del directorio, explicando el propósito de cada archivo dentro del repositorio. Finalmente, se proporciona una guía para la ejecución del microservicio de forma local, facilitando pruebas y desarrollos adicionales sin necesidad de un entorno productivo. Esta documentación garantiza la comprensión, mantenibilidad y escalabilidad del microservicio, asegurando que cualquier usuario o desarrollador pueda integrarse fácilmente al proyecto.

4.8. Solicitud de un *Pull Request* a la Rama *Master*

La rama *Master* representa el entorno productivo, donde los programas generan valor y son esenciales para la operación de la empresa. Para que un *software* alcance esta etapa, primero debe ser integrado en la rama de desarrollo, luego en la rama de QA y, finalmente, en la rama *Master*, tras haber cumplido con los requisitos y estándares establecidos por Grupo Éxito. En este caso, el microservicio desarrollado inició este proceso durante los primeros días de enero de 2025 y, en un período de cinco días hábiles, fue aprobado e integrado en la rama *Master*. Esto permitió que su flujo de trabajo se activara automáticamente, asegurando su correcta ejecución en el entorno productivo mediante OKS (*On-Premise Kubernetes*). La generación del *Pull Request* a la rama *Master* garantiza que el microservicio se despliegue de manera eficiente y confiable, contribuyendo a la estabilidad y continuidad operativa del sistema.

4.9. Tablero de Monitoreo de MIPS en el Entorno Productivo

Para que el producto final, un tablero de *Power BI* que integra los análisis y transformaciones de datos realizados por la aplicación de *Python*, sea accesible para el Equipo de Gestión de MIPS y el Equipo de Operaciones, es necesario publicarlo en la versión *Online* de *Power BI* y gestionar los permisos de acceso. Para ello, se debe realizar una solicitud al encargado de la administración de tableros y permisos, adjuntando el archivo del tablero junto con el listado de personas que tendrán acceso. Este proceso se llevó a cabo en las últimas etapas del proyecto y tomó solo unas pocas horas, asegurando que los equipos clave pudieran visualizar la información de manera oportuna y eficiente.

4.10. Documentación del Proyecto

Para finalizar, se elaboró la documentación del proyecto, la cual detalla tanto la justificación como la metodología utilizada en su desarrollo. En este documento se expone el propósito del proyecto, resaltando la importancia de su implementación y los beneficios esperados. Asimismo, se describen los fundamentos teóricos en los que se sustenta, proporcionando un marco conceptual sólido para comprender su funcionamiento. Esta documentación garantiza la trazabilidad del proyecto y sirve como referencia para futuras mejoras o ampliaciones.

5. Conclusiones

El desarrollo del presente trabajo permitió analizar en profundidad el consumo de MIPS, la ejecución de procesos y la dinámica operativa de la compañía. A través del análisis exploratorio de datos, se identificaron patrones clave, estableciendo que un subconjunto reducido de procesos es responsable de la mayor parte del consumo de recursos. Se evidenció también una marcada periodicidad en el uso de MIPS, con mayor concentración en días hábiles y una reducción significativa en fines de semana y festivos, reflejando el ciclo laboral de la empresa.

La normalización de la tabla `refrescarprocesos_10dias` representó un avance significativo en la gestión de datos, reduciendo la redundancia y mejorando la integridad de la información. Esta reestructuración facilitó la escalabilidad del sistema y optimizó la eficiencia en la carga de datos, beneficiando la construcción de tableros interactivos en *Power BI*.

El diseño del flujo de trabajo del microservicio permitió la extracción, procesamiento y almacenamiento de datos de manera estructurada. Mediante una aplicación en *Python*, se implementaron tareas de normalización, detección de anomalías y predicción de consumo de MIPS, cuyos resultados fueron almacenados en una base de datos destino y visualizados en *Power BI*. Para la predicción del consumo de MIPS, se desarrollaron modelos basados en la biblioteca `PROPHET`, mientras que para la detección de anomalías se diseñó un modelo que empleó el Rango Intercuartílico (IQR) y la Desviación Absoluta Mediana (MAD).

El tablero de *Power BI* construido proporciona una herramienta integral para el monitoreo y optimización del consumo de MIPS. Sus secciones permiten analizar datos generales, detectar procesos anómalos, examinar variables explicativas y evaluar métricas de precisión del modelo predictivo. La publicación del tablero en la versión *online* y la gestión de permisos garantizan que los equipos clave puedan acceder a la información de manera oportuna y eficiente.

Para garantizar un despliegue automatizado y seguro, se configuró un *pipeline* en *Azure DevOps* y se implementó la arquitectura en *Kubernetes* mediante *Helm* y *Docker*. La integración del microservicio en la rama *Master* aseguró su correcta ejecución en el entorno productivo mediante *OKS*, permitiendo un despliegue confiable que contribuye a la estabilidad y continuidad operativa del sistema.

El objetivo del microservicio se cumplió en su totalidad, logrando la automatización de los reportes diarios, la detección de procesos con consumos atípicos y la predicción del consumo de MIPS con un nivel de precisión adecuado. Sin embargo, aunque las métricas MAE, MAPE y sMAPE lograron mantenerse por debajo del umbral del 15 % establecido, la métrica RMSE no logró estabilizarse dentro de este límite. Esto se debe a que el RMSE penaliza fuertemente las desviaciones grandes debido al uso del cuadrado de la diferencia (Ver Subsección 2.7.1), lo que amplifica el impacto de valores extremos en el error total. A pesar de ello, el modelo predictivo logró capturar de manera efectiva las estacionalidades semanales y anuales, proporcionando una visión razonable del comportamiento del consumo de MIPS.

Finalmente, la documentación completa del proyecto proporciona un marco sólido para su comprensión, mantenimiento y escalabilidad. En ella se detallan la justificación, metodología y fundamentos teóricos, garantizando la trazabilidad del sistema y sirviendo como referencia para futuras mejoras o ampliaciones. Para trabajos futuros, se recomienda realizar un análisis del consumo de MIPS que integre el efecto combinado del número total de ejecuciones y el número de registros, lo que podría aportar información relevante para una mejor comprensión del comportamiento operativo y la eficiencia en el uso de los recursos tecnológicos.

Referencias

[1] Charu C. Aggarwal, *Outlier analysis*, 2 ed., Springer, Cham, 2017, Copyright Springer International Publishing AG 2017. Hardcover published 2016, Softcover published 2018.

[2] Mehala Balamurali and Raymond Leung, *Statistical outliers*, pp. 1443–1451, Springer International Publishing, Cham, 2023.

[3] Sabyasachi Basu and Martin Meckesheimer, *Automatic outlier detection for time series: an application to sensor data*, Knowledge and Information Systems **11** (2007), no. 2, 137–154.

[4] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano, *A review on outlier/anomaly detection in time series data*, ACM (2020), Manuscript submitted to ACM.

[5] D. Chicco, M. J. Warrens, and G. Jurman, *The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation*, PeerJ Computer Science **7** (2021), e623.

[6] Everette S Gardner Jr, *Exponential smoothing: The state of the art*, Journal of forecasting **4** (1985), no. 1, 1–28.

[7] Andrew Harvey and Simon Peters, *Estimation procedures for structural time series models*, Journal of Forecasting **9** (1990), 89–108.

[8] Trevor Hastie and Robert Tibshirani, *Generalized additive models; some applications*, Generalized Linear Models (New York, NY) (Robert Gilchrist, Brian Francis, and Joe Whittaker, eds.), Springer US, 1985, pp. 66–81.

[9] D. M. Hawkins, *Identification of outliers*, 1 ed., Monographs on Statistics and Applied Probability, Springer Dordrecht, 1980, Copyright D. M. Hawkins 1980. Softcover published 2014, eBook published 2013.

[10] Matthieu Komorowski, Dominic C. Marshall, Justin D. Salciccioli, and Yves Crutain, *Exploratory data analysis*, pp. 185–203, Springer International Publishing, Cham, 2016.

[11] Julia Martins, *Scrum: conceptos clave y cómo se aplica en la gestión de proyectos*, Blog de Asana, febrero 2025, Consultado el 4 de marzo de 2025.

[12] Z. Qu, W. Dai, C. Euan, et al., *Exploratory functional data analysis*, TEST (2024).

[13] Dan Radigan, *Backlog del producto: consejos para crear y priorizar*, Blog de Atlassian, 2025, Consultado el 7 de febrero de 2025.

[14] D. Rindskopf and M. Shiyko, *Measures of dispersion, skewness and kurtosis*, International Encyclopedia of Education (Third Edition) (Penelope Peterson, Eva Baker, and Barry McGaw, eds.), Elsevier, Oxford, third edition ed., 2010, pp. 267–273.

[15] Ken Schwaber and Jeff Sutherland, *La guía scrum: La guía definitiva de scrum: Las reglas del juego*, November 2020, © 2020 Ken Schwaber and Jeff Sutherland. Disponible bajo la licencia Creative Commons Attribution Share-Alike.

[16] Howard J. Seltman, *Experimental design and analysis*, `http://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf`, 2012, Online resource accessed on January 25, 2025.

[17] Facebook Open Source, *Prophet: Forecasting at scale*, 2023, Accessed: 2024-11-12.

[18] Sean J. Taylor and Benjamin Letham, *Forecasting at scale*, PeerJ Preprints **5** (2017), e3190v2.

[19] John W Tukey, *Exploratory data analysis*, Reading/Addison-Wesley (1977).

[20] Unisys, *Clearpath forward*, 2025, Último acceso: 9 de enero de 2025.

[21] Rand R. Wilcox, *3 - summarizing data*, Applying Contemporary Statistical Techniques (Rand R. Wilcox, ed.), Academic Press, Burlington, 2003, pp. 55–91.