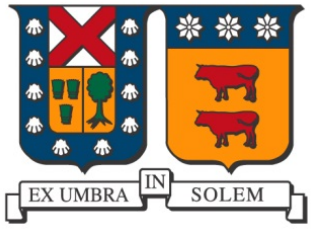
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE




# “DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN ÁGIL PARA EL ANÁLISIS Y SEGUIMIENTO DE DATOS EN LAS VENTANAS DE DESPACHO CASO PRÁCTICO: WALMART CHILE”

JAVIERA ROJAS GARCÍA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN INFORMÁTICA

Profesor Guía: Marcello Visconti
Profesor Correferente: Francisco Cabezas

Agosto - 2023

# DEDICATORIA

Dedico este trabajo a las personas que me han apoyado en cada paso de este viaje.

A mis queridos padres, quienes me han enseñado con su ejemplo el valor de la persistencia y la dedicación. En especial a mi madre, cuyo amor, apoyo incondicional y sabiduría han sido mi faro en los momentos más oscuros. Gracias por creer en mí incluso cuando dudaba de mí misma.

A mi hermana, cuya fortaleza y carácter siempre me han inspirado. Y a mi pequeña sobrina, que con su alegría y entusiasmo me ha recordado constantemente el valor de la curiosidad y lo maravilloso que es vivir en este mundo.

A mi abuelito, espero que si existe algo más allá de esta vida, puedas ver este logro y sentirte orgulloso. Y a mi abuelita, cuyo recuerdo vive a pesar de que su mente nos dejó antes de tiempo, siempre valoraré los años en que me amaste con todo tu corazón.

A mis amigos, por ser mi sistema de apoyo y mis compañeros de aventuras a través de los años. En especial a Paula, que ha estado conmigo desde el día uno, compartiendo risas y lágrimas, éxitos y fracasos. A Pablo, contigo confirmé que las almas gemelas viven en las amistades. Su amistad ha sido un recordatorio constante de que no estoy sola en esto.

Por último, a mis compañeros de cuatro patas, a Luke, Artoo y Kiara. Aunque no pueden entender las palabras en estas páginas, su compañía incondicional y cariño han sido un bálsamo en los días más difíciles, a su manera única, han aliviado el estrés de las noches más largas y ha enriquecido los momentos de descanso.

Esta tesis es el resultado de nuestra travesía compartida y es a todos ustedes a quienes se la dedico con todo mi cariño y gratitud.

# AGRADECIMIENTOS

Primero que nada, quiero expresar mi más profundo agradecimiento a mi madre. Su amor, su valor para dejarme seguir mi propio camino, y su apoyo incondicional en todo momento han sido el pilar que me ha mantenido en pie. Cada caída ha sido una lección, y su ejemplo me ha enseñado a levantarme cada vez.

A mi hermana, que nunca dejó de recordarme la importancia de escribir esta memoria. Su constante ”molestia” fue una motivación invaluable y siempre la recordaré con cariño, y a mi padre, cuyo apoyo ha sido fundamental en este viaje.

Mi agradecimiento también a Francisco Cabezas, mi mentor en Walmart, quien no sólo me proporcionó todas las herramientas necesarias para terminar este trabajo, sino que también me alentó hasta el último día a ser mejor profesional. De paso no olvido agradecer al equipo Kronos, quienes me hacen sentir una parte integral del equipo.

Deseo agradecer a los profesores del departamento de informática, cuyo entusiasmo y pasión por la enseñanza mantuvieron vivo mi deseo de aprender. Sin su orientación y apoyo, no habría podido llegar a este punto en mi carrera académica y pensar en terminar este proceso.

A mis amigos, que siempre han estado ahí para mí, incluso cuando me pierdo en mi trabajo. Gracias por recordarme siempre la importancia de finalizar este proceso. Paula, Pablo, Alexis, Cata, gracias por soportar mi estrés universitario y por estar conmigo cada vez que lo necesité, sin duda hicieron de Valparaíso un hogar. A mis amigos de informática, gracias por aceptarme y hacerme parte de sus vidas, definitivamente nunca me sentí excluida con ustedes. A mis amigas del colegio, aunque la distancia nos separaba, su apoyo siempre estuvo presente. Y a Alito, cuya amistad ha resistido el paso del tiempo y las diversas etapas de la vida, gracias por ser mi cómplice desde niños.

A todos ustedes, gracias. Este logro es tan mío como de ustedes.

# RESUMEN

**Resumen—** En un mundo impulsado por la información, donde la toma de decisiones basada en datos puede marcar la diferencia entre el éxito y el fracaso, la optimización de los servicios se ha convertido en un imperativo. En este estudio, se propone la incorporación de un sistema de análisis de datos en Walmart CL, específicamente en ”Promise Engine”, el artefacto encargado de las ventanas de despacho. Se enfatiza la utilización exclusiva de tecnologías avaladas por la empresa y la implementación bajo metodologías ágiles, alineándose con la cultura corporativa de Walmart. Durante el desarrollo, se enfrentó el reto adicional de un proceso de migración de la plataforma de Chile a Internacional. Los resultados obtenidos muestran un flujo de datos efectivo desde la API del ”Promise Engine”, hacia un datamart en BigQuery, todo ello monitoreado. Basándose en hipótesis sobre el comportamiento de las ventanas de despacho, se crearon diversos dashboards para su validación. Las conclusiones destacan la importancia de seguir el flujo de trabajo empresarial, a pesar de los desafíos, y la adaptabilidad en el desarrollo de soluciones.

**Palabras Clave—**

Ingeniería de Datos, Ventanas de Despacho, Inteligencia de Negocios, Metodología Ágil.

# ABSTRACT

***Abstract*—** In a world driven by information, where data-based decision-making can make the difference between success and failure, optimizing services has become imperative. In this study, the incorporation of a data analysis system is proposed for Walmart CL, specifically in the ”Promise Engine”, the tool responsible for the dispatch windows. Emphasis is placed on the exclusive use of technologies endorsed by the company and the implementation under agile methodologies, aligning with Walmart’s corporate culture. During the development, an additional challenge was faced with migrating the platform from Chile to an International one. The results showcase an effective data flow from the ”Promise Engine” API to a datamart in BigQuery, all being monitored. Based on hypotheses regarding the behavior of the dispatch windows, various dashboards were created for their validation. The conclusions emphasize the importance of adhering to the corporate workflow, despite challenges, and the adaptability in solution development.

***Keywords*—** Data Engineering, Slots, Business Intelligence, Agile Methodology.

# GLOSARIO

**BI**: Business Intelligence - Inteligencia de negocio.
**CD**: Centro de Distribución.
**Centralized**: Tipo de inventario en el que un tercero vende su producto a través de Líder y lo lleva a los CD para su envío.
**CL**: Chile - Refiere al ecosistema de Walmart en Chile, no a nivel internacional.
**Delivery**: Envío.
**DI**: Departamento de Informática.
**DM**: Datamart - Subconjunto de un datawarehouse orientado a un departamento o función específica.
**DW**: Datawarehouse - Base de datos centralizada para consolidación, análisis y reporte de datos.
**GM**: Venta de catálogo extendido.
**GR**: Grocery - Venta de supermercado.
**GTP**: Global Tech Platform - Plataforma en la red interna de Walmart International.
**HD**: Home Delivery - Modalidad de envío a domicilio.
**KITT**: Kubernetes In The Trenches - Herramienta de Walmart para la transición a la nube.
**KPI**: Key Performance Indicator - Indicador clave de rendimiento (aunque no especificaste su traducción, esta es la más común).
**Lead time**: Tiempo de espera, definido según el contexto.
**Next Day Delivery**: Envío del paquete al día siguiente de la compra.
**OnHand**: Tipo de Inventario donde Líder vende el producto.
**Picking**: Proceso de búsqueda del paquete en el centro de distribución.
**POD**: Instancia de un proceso en un clúster de Kubernetes.
**S2S**: Ship to Store - Modalidad de retiro en tienda.
**Same Day Picking**: Picking del paquete el mismo día de su despacho.
**Shift**: Ventana de picking, momento en que se realiza el picking de una orden.
**Slot**: Ventana de despacho.
**US**: United States - En este contexto, Walmart US o Walmart Internacional.
**UTFSM**: Universidad Técnica Federico Santa María.
**Whitelist**: Lista blanca, elementos permitidos.

# ÍNDICE DE CONTENIDOS

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS

# INTRODUCCIÓN

En la era contemporánea, las organizaciones empresariales buscan de manera constante mejorar la calidad y eficacia de los productos o servicios que proporcionan a sus clientes. En este contexto, se ha desarrollado un proyecto de integración de un sistema de análisis y seguimiento de datos en Walmart CL, centrado en la gestión de las ventanas de despacho. Esta iniciativa tiene por objetivo alcanzar objetivos cruciales como la mejora de productos y/o servicios, la maximización del retorno económico, y la minimización del riesgo asociado a la toma de decisiones.

Esta tesis se centra en la propuesta de un sistema que permita un análisis de datos detallado y un seguimiento continuo, con la capacidad de generar un flujo de datos que se extiende desde el componente responsable de los eventos relacionados con las ventanas de despacho hasta un depósito de datos o datamart. A partir de aquí, se pueden desarrollar soluciones para la visualización de datos y análisis adicionales. Este proceso se lleva a cabo con un estricto cumplimiento de las tecnologías autorizadas por Walmart Tech, asegurando así una implementación efectiva y segura en el entorno empresarial existente.

# CAPÍTULO 1
# DEFINICIÓN DEL PROBLEMA

## 1.1. Contexto preliminar

En Walmart Chile se han implementado diversas iniciativas centradas en una estrategia de datos emergente. Esta estrategia tiene como objetivo promover la toma de decisiones basadas en datos entre los equipos. En paralelo, la empresa está pasando por un proceso de migración significativo en el que todos los productos deben ser transferidos a la plataforma global de Walmart International (GTP). Esta transición implica el uso exclusivo de tecnologías aprobadas por la matriz internacional.

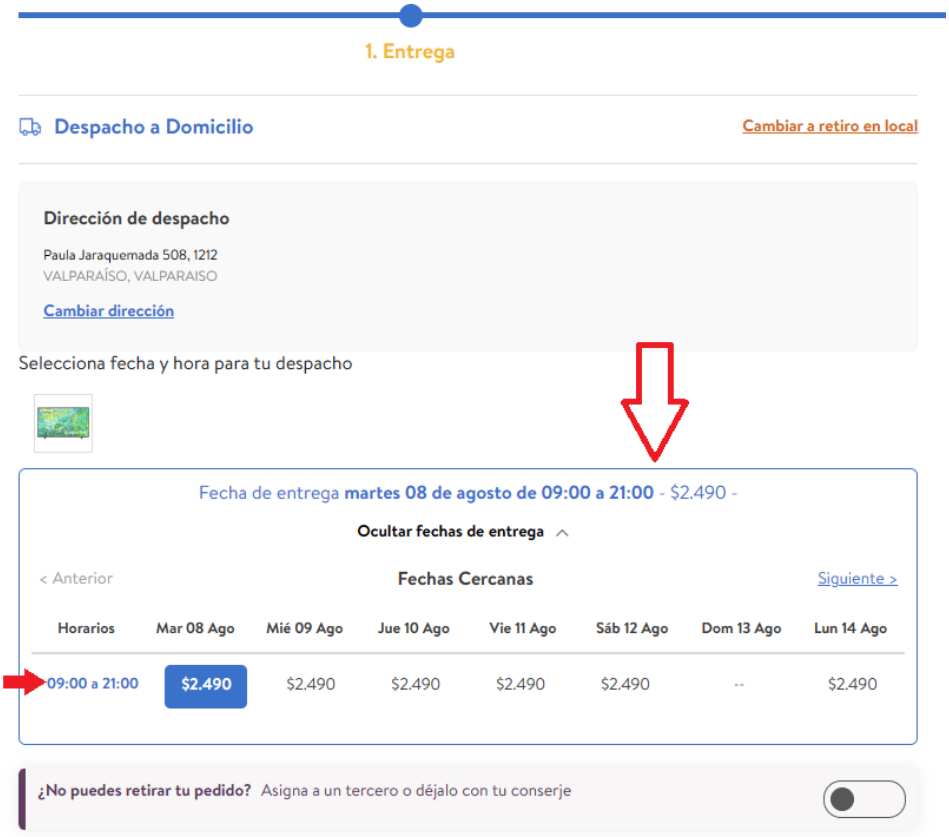



Figura 1: Oferta de Ventanas en Lider CL
Fuente: Elaboración Propia.

El ”Promise Engine” es el sistema back-end responsable de administrar las ventanas de entrega en el e-commerce de la cadena de supermercados Líder. Este sistema opera en tres

etapas fundamentales:

1. **Oferta de ventanas**: Esta es la primera interacción del cliente con el sistema, donde se le presentan todas las ventanas de entrega disponibles para su pedido. Este proceso ocurre antes de realizar el pago durante el proceso de compra, en la figura 1 se puede apreciar en qué parte del flujo en la web se ofrecen las ventanas de despacho, se puede ver que se ofrecen múltiples ventanas por cada oferta.
2. **Reserva de ventanas**: Ocurre cuando el cliente selecciona una ventana de entrega y procede con la compra. Este paso también se realiza antes de completar el pago en el proceso de compra, pero después de hacer click en ”pagar”.
3. **Confirmación de ventanas**: Se realiza una vez que el cliente ha seleccionado la ventana de entrega y ha procedido a realizar la compra. Esto es una vez se efectúa el pago en el proceso de compra.

El ”Promise Engine”, por su naturaleza, es altamente dependiente de la logística, lo cual lo hace reactivo a ésta. En este contexto, el desafío es lograr un equilibrio que también tenga en cuenta al cliente y su comportamiento. De este modo, se estaría alineando con la nueva estrategia de datos de la empresa.

## 1.2. Identificación del problema

En la era moderna, el análisis de datos juega un papel crucial en las organizaciones, permitiéndoles abrirse a nuevas oportunidades y acercarse a una gestión más efectiva. Algunos de los beneficios clave incluyen:

- Mejora del servicio al cliente
- Predicción de tendencias de ventas
- Mejora en la toma de decisiones
- Resolución eficaz de problemas
- Aumento de la base de clientes
- Supervisión del rendimiento y mejora de los procesos
- Entendimiento del comportamiento del cliente y el mercado

Actualmente, el Promise Engine no cuenta con un sistema que permita el análisis y seguimiento de los datos. Dado el énfasis contemporáneo en la toma de decisiones basada en datos, se hace evidente la necesidad de implementar una estrategia de este tipo, reduciendo la dependencia de la intuición humana y facilitando la comprensión de por qué los clientes abandonan el proceso de compra en la ventana de despacho.

La solución de este problema facilitaría la toma de decisiones informadas sobre el desarrollo futuro del producto, proporcionaría métricas actualizadas sobre el proceso de oferta, reserva y confirmación de ventanas, y permitiría una comprensión más profunda del comportamiento del cliente.

## 1.3. Objetivos de la memoria

### 1.3.1. Objetivo General

Diseñar e implementar un sistema para el análisis y seguimiento de los datos relacionados con las ventanas de despacho para el canal catálogo extendido, todo esto utilizando una metodología ágil para adaptarse rápidamente a las cambiantes necesidades del mundo empresarial.

### 1.3.2. Objetivos Específicos

- Plantear hipótesis en el contexto de las ventanas de despacho
- Diseñar una arquitectura de almacenamiento de datos
- Realizar la migración de los datos existentes
- Establecer el pipeline de datos
- Crear dashboards con orientación al negocio
- Validar las hipótesis planteadas

# CAPÍTULO 2
# MARCO CONCEPTUAL

## 2.1. Metodología Ágil

En los comienzos del desarrollo de software, la mayoría de los requerimientos eran bastante estables y los planes se ejecutaban sin grandes cambios en el proceso. Hoy en día eso ha cambiado drásticamente, el desarrollo de software tiene el desafío de crear proyectos dinámicos, donde van surgiendo más dificultades y requerimientos a medida que el mercado cambia y las empresas crecen. Algunas de estas dificultades involucran [Kent Beck, 2009] [D. Turk y B, 2002] :

- **La evolución de los requerimientos:** Los requerimientos de los clientes y empresas evolucionan constantemente dada las necesidades del negocio o incluso cambios legislativos. La mayoría de los clientes no tienen claras todas las especificaciones de los requerimientos al comienzo del desarrollo.
- **Intervención del cliente:** Si el cliente no se involucra en el desarrollo se tienen mayores chances de que el proyecto falle, previamente muchas compañías no se esforzaban en incluir al cliente.
- **Fechas límites y presupuesto:** Muchas compañías no ofrecen el suficiente presupuesto y tiempo pero aún así requieren de implementar features altamente demandantes y todo esto por la competencia del mercado.
- **Falta de comunicación:** Por último pero no menos importante es la principal causa de malentendido de los requerimientos, este es la mala comunicación entre la operación y el cliente, especialmente si no se está hablando en el mismo lenguaje (refiriéndose a que quizás una parte esté enfocado solo en la parte técnica y la otra solo maneje el lenguaje de cara al negocio).

Por esto mismo que el agilismo llegó a ocupar un rol tan importante en el desarrollo, y están basadas en la regla de que la mejor forma de desarrollar un sistema es la entrega constante de versiones funcionales al cliente, y a partir de eso hacerles actualizaciones de acuerdo al feedback.

Además de esto contempla aceptar cambios de requerimientos en cualquier momento del flujo de desarrollo, así los clientes se sentirán cómodos en y ayuda en el siguiente punto, se debe tener constante cooperación entre los desarrolladores y clientes en todo el desarrollo del proyecto, y por último es tener un desarrollo guiado por pruebas, lo que significa que por cada código escrito se tendrá test que lo prueben.

El agilismo es una metodología utilizada desde hace años, que se utiliza en organizaciones de TI para crear software o gestionar procesos de forma más eficaz. En términos generales, es un enfoque colaborativo en el que los equipos multifuncionales diseñan y construyen productos mínimos viables (MVP) y funcionalidades rápidamente, los prueban con los clientes y los perfeccionan y mejoran en iteraciones rápidas[Santiago Comella-Dorda y Speksnijder, 2016]

Algunas empresas están usando el agilismo en sus estrategias de datos. Agile Data tiene los mismos fundamentos que el uso del agilismo en el desarrollo de software, se tienen equipos funcionales que incluye tanto el conocimiento del negocio e IT para trabajar en data labs, estos están enfocados en generar insights confiables que permitan a la compañía obtener resultados rápidos y tomar los requerimientos de mayor prioridad respecto al negocio. [Chiara Brocchi y Neiman, 2016]

## 2.2. Introducción al Scrumban

Scrumban integra prácticas de Scrum y Kanban en un enfoque ágil unificado. Surgió como un camino para facilitar la transición de los equipos de Scrum a Kanban, pero ha evolucionado hasta convertirse en una práctica autónoma con su propio conjunto de principios y prácticas. Scrumban proporciona la estructura iterativa de Scrum y la visualización y eficiencia del flujo de trabajo de Kanban, lo que permite a los equipos adaptarse rápidamente a los cambios sin sacrificar la calidad o la eficiencia. En Scrumban, las historias de usuario y los spikes (tareas que no producen código ejecutable pero que ayudan a explorar soluciones o reducir la incertidumbre) son utilizados para gestionar y planificar el trabajo [Ladas, 2009].

### 2.2.1. Uso de Scrumban

Scrumban es altamente adaptable y puede ser utilizado en una variedad de proyectos y entornos, pero es particularmente útil en entornos de desarrollo que cambian rápidamente y requieren una gran flexibilidad y adaptabilidad. En Scrumban, los equipos trabajan en un flujo de trabajo pull, donde los miembros del equipo ”pull.º toman las tareas de una pila de ”para hacerz las mueven a través de una serie de estados hasta que se completan. Esto da a los equipos una gran flexibilidad para adaptarse a los cambios de prioridades [Banguero y Amaya, 2018]. Scrumban también promueve la mejora continua, permitiendo a los equipos iterar y mejorar sus procesos con el tiempo.

## 2.3. Comparación con Scrum y Kanban

### 2.3.1. Diferencias con Scrum y Kanban

Aunque Scrumban comparte muchas similitudes con Scrum, hay algunas diferencias clave. A diferencia de Scrum, que se basa en sprints fijos y tiene roles claramente definidos (como el dueño del producto y el scrum master), Scrumban no tiene una estructura de tiempo fija y los roles son menos prescriptivos. Esto da a los equipos una mayor flexibilidad para adaptarse a los cambios a medida que ocurren [Sutherland, 2014].

Kanban y Scrumban comparten el enfoque en la visualización del flujo de trabajo y la eficiencia en el flujo de trabajo. Sin embargo, a diferencia de Kanban, que es un sistema puramente pull, Scrumban introduce iteraciones, eventos de revisión y planificación, y otras prácticas de Scrum. Kanban se centra en minimizar los cuellos de botella y maximizar el flujo de trabajo, pero no tiene eventos de tiempo-box como las reuniones de planificación y las retrospectivas que se encuentran en Scrumban y Scrum [Anderson, 2010].

Scrumban ofrece una combinación efectiva de Scrum y Kanban, proporcionando la estructura iterativa de Scrum y la flexibilidad de Kanban. Esta combinación permite a los equipos equilibrar eficazmente la previsibilidad y la adaptabilidad en sus procesos de desarrollo de software. El uso efectivo de Scrumban requiere un compromiso con los principios ágiles, una comprensión clara de las historias de usuario y spikes, y una disposición para experimentar y aprender constantemente [Ladas, 2009].

## 2.4. Business Intelligence

La Inteligencia de Negocios, o Business Intelligence (BI), no es un producto de desarrollos en el campo de las ciencias administrativas, sino que es un resultado del avance de la informática y de la ”infotecnología”[Negash, 2004]. La BI tiene sus raíces en el creciente desafío de manejar la gran cantidad de datos que las empresas generan y recopilan.

Las organizaciones utilizan los sistemas de BI para transformar estos datos en información valiosa, con el objetivo de mejorar el proceso de toma de decisiones [Ranjan, 2008]. Estos sistemas permiten identificar, almacenar, analizar y generar informes sobre la información del negocio. En particular, la BI se usa para identificar patrones y tendencias que pueden ayudar a las organizaciones a comprender su rendimiento, la competencia y el mercado [Davenport y Harris, 2006].

El poder de la BI radica en su capacidad para convertir una gran cantidad de datos en información procesable. Los sistemas de BI son valiosos porque proporcionan acceso a datos de la empresa, los analizan y los presentan de tal manera que los tomadores de decisiones pueden contar con información más procesada, permitiendo así una organización basada en

evidencias [Watson, 2010]. Esta capacidad para informar las decisiones basándose en datos en lugar de en intuiciones es lo que hace que la BI sea un activo tan valioso para las empresas.

En resumen, la información resultante del uso de los sistemas de BI se considera un activo empresarial debido a los beneficios que aporta su uso. A través de la BI, las organizaciones pueden tomar decisiones basadas en datos, mejorar la eficiencia, identificar nuevas oportunidades de negocio y obtener una ventaja competitiva [Chugh y Grandhi, 2013].

## 2.5. Ingeniería de Datos

La Ingeniería de Datos es una disciplina esencial en el ámbito de la ciencia de datos, centrada en el diseño, construcción y gestión de soluciones que permiten el manejo adecuado de la información a gran escala. En la era actual, caracterizada por la masiva generación de datos, la figura del ingeniero de datos ha adquirido una importancia sin precedentes, asegurando que los datos estén estructurados, accesibles y listos para su análisis [Zikopoulos y Eaton, 2012].

Las responsabilidades fundamentales de la Ingeniería de Datos incluyen:

- **Diseño de sistemas de almacenamiento:** Crear y mantener sistemas como bases de datos y lagos de datos que satisfagan las necesidades de procesamiento y análisis de la organización.
- **Procesos ETL (Extract, Transform, Load):** Esenciales para trasladar y transformar datos entre sistemas y asegurar su correcta carga en plataformas analíticas [Kimball y Ross, 2008].
- **Optimización de rendimiento:** Garantizar que los sistemas de datos operen eficientemente, adaptándose a cargas de trabajo cambiantes.
- **Seguridad de datos:** Proteger los datos contra accesos no autorizados y cumplir con normativas pertinentes.
- **Integración y colaboración:** Conectar diferentes sistemas y fuentes de datos para proporcionar una perspectiva coherente de la información en la organización.

La conexión con la Business Intelligence (BI) es intrínseca. Mientras que la BI se centra en interpretar y visualizar datos para apoyar la toma de decisiones, la Ingeniería de Datos garantiza que la infraestructura y herramientas necesarias estén disponibles para alimentar esos análisis. Sin la base sólida que ofrece la Ingeniería de Datos, las soluciones de BI no serían eficientes. En resumen, los ingenieros de datos establecen las bases para que los profesionales de BI puedan extraer valiosos insights [Marr, 2015].

## 2.6. Data Mart y Data Warehousing

En el contexto de la Inteligencia de Negocios, la construcción y diseño de almacenes de datos, conocidos como Data Warehouses (DW) o Data Marts (DM), desempeña un papel vital [y M. Garcia, 2017]. Un DW es una base de datos corporativa que consolida datos provenientes de diversos sistemas dentro de una organización, proporcionando una visión unificada y coherente de los datos corporativos.

Por otro lado, un DM es una versión más pequeña y enfocada de un DW [y M. Garcia, 2017]. Está diseñado para satisfacer las necesidades de un departamento o área funcional específica dentro de la organización. En otras palabras, un DM es una implementación de DW a una escala más restringida, especializada en el almacenamiento de datos relacionados con un área de negocio en particular.

Los DW y DM se estructuran lógicamente en tablas de dimensiones y de hechos. Las tablas de dimensiones albergan los conceptos del negocio que se van a analizar, mientras que las tablas de hechos contienen las medidas o indicadores del negocio. Esta organización facilita la creación de cubos multidimensionales y el procesamiento analítico en línea [Joshua y Ratnam, 2018].

La principal función de los DW y DM es apoyar el proceso de toma de decisiones, no realizar transacciones. Al proporcionar un acceso rápido y eficiente a los datos relevantes para las decisiones estratégicas, estos sistemas ayudan a las organizaciones a ser más ágiles y basadas en datos [Joshua y Ratnam, 2018].

Además, DW y DM permiten una visión histórica de los datos, lo que facilita la identificación de tendencias y patrones a lo largo del tiempo. Esta capacidad para soportar análisis temporales es otro factor que los hace valiosos para las decisiones estratégicas.

### 2.6.1. BigQuery

Google BigQuery emerge como una solución vanguardista en el ámbito del análisis de datos, especialmente por su naturaleza altamente escalable y la capacidad de manejar inmensos volúmenes de información. Como delinean Jethani et al. [Jethani *et al.*, 2021], esta plataforma es una propuesta de ”servicio de análisis de datos económico y con una alta capacidad de escalabilidad”, aprovechando una arquitectura de procesamiento distribuido para acelerar la obtención de resultados en las consultas de datos. Esta característica es esencial, ya que permite a los profesionales realizar investigaciones SQL avanzadas en conjuntos de datos colosales, proporcionando insights prácticamente en tiempo real.

Paralelamente, uno de los rasgos más distintivos y provechosos de BigQuery es su habilidad para implementar ”vistas”. Estas son, esencialmente, consultas SQL preestablecidas que operan como tablas virtuales. Aunque no almacenan información intrínsecamente, sirven

como referencia para la consulta que simbolizan. El verdadero valor de las vistas reside en su capacidad para acelerar el proceso de consulta, en especial cuando se necesita acceder con frecuencia a ciertos conjuntos de datos que demandan transformaciones particulares [López, 2022]. Estas vistas predefinidas suprimen la necesidad de generar múltiples tablas físicas, resultando en un ahorro significativo en términos de espacio y recursos. Asimismo, encapsulan la lógica detrás de una consulta, minimizando las probabilidades de errores en análisis subsiguientes. Al integrarse con el mecanismo de caché de BigQuery, este enfoque potencia el rendimiento y posibilita la obtención ágil y precisa de información valiosa.

En complemento a sus robustas capacidades de análisis, BigQuery brinda servicios de almacenamiento en la nube y gestión de datos de gran solidez. Su sinergia con otras herramientas de la suite de Google Cloud Platform promueve un ambiente integral para el procesamiento, estudio y visualización de datos, consolidando a BigQuery como la elección óptima para las demandas contemporáneas en análisis de datos en la nube [Jethani *et al.*, 2021].

### 2.6.2. Apache Kafka

Apache Kafka es un sistema de mensajería distribuido y de alta capacidad, diseñado para permitir el manejo eficiente de grandes volúmenes de datos y facilitar la comunicación entre sistemas mediante un modelo de publicación y suscripción [Desai y Guy, 2019]. Este poderoso sistema de streaming de eventos ha sido diseñado con la intención de proporcionar una plataforma unificada, de alta capacidad y baja latencia para manejar en tiempo real el flujo de datos generados por fuentes diversas.

Entre los beneficios más significativos que ofrece Apache Kafka, se incluyen:

- **Fiabilidad**: Gracias a su diseño distribuido, con datos particionados y replicados, Kafka proporciona alta tolerancia a fallos y garantiza la durabilidad de los datos.
- **Escalabilidad**: Kafka puede crecer junto con las necesidades de procesamiento de datos, permitiendo una escalabilidad sin tiempo de inactividad.
- **Durabilidad**: Los mensajes en Kafka son persistentes en el disco y se mantienen durante el tiempo que sea necesario para asegurar su consumo.
- **Rendimiento**: Kafka ofrece un alto rendimiento tanto en la publicación como en la suscripción de mensajes, manteniendo un buen rendimiento incluso con terabytes de datos almacenados.

Además, Kafka puede soportar la entrega de mensajes con baja latencia y garantizar la tolerancia a fallos en presencia de fallos de máquina. Tiene la capacidad de manejar un gran número de consumidores diferentes y puede realizar hasta dos millones de escrituras por segundo [Desai y Guy, 2019].

### 2.6.3. Tópicos en Apache Kafka

En Apache Kafka, un concepto esencial es el de los tópicos. Un tópico es una categoría o flujo de datos a la que los productores envían y de la cual los consumidores reciben los mensajes [Kleppmann, 2017]. En un esquema típico, los productores escriben datos en los tópicos y los consumidores leen de los tópicos. Los tópicos en Kafka son multi-suscriptor, es decir, un tópico puede tener cero, uno o muchos consumidores que se suscriben a los datos escritos en él [Kleppmann, 2017].

Desde el punto de vista del almacenamiento, un tópico se divide en particiones, cada una de las cuales mantiene un orden de los mensajes que garantiza que los datos se consumen en el orden en que fueron producidos. Además, Kafka replica cada partición en varios nodos para garantizar la redundancia de los datos y permitir el acceso a los mismos en caso de fallo de un nodo [Kleppmann, 2017].

Un aspecto poderoso de los tópicos en Kafka es que permiten a los usuarios segmentar sus datos. Los productores pueden escribir datos en tópicos específicos, y los consumidores pueden leer de tópicos específicos. Esto permite que los datos se categoricen de manera efectiva según las necesidades de la aplicación [Kleppmann, 2017].

De este modo, los tópicos en Kafka desempeñan un papel fundamental en la organización y el acceso a los flujos de datos en tiempo real, proporcionando la infraestructura necesaria para el procesamiento eficiente de grandes volúmenes de datos en una variedad de aplicaciones de análisis de datos y de streaming.

### 2.6.4. Schema Avro y su Relación con Apache Kafka

El Schema Avro es una especificación formal para la estructura de los datos que se utilizarán en la serialización y deserialización en sistemas distribuidos. Se define utilizando una sintaxis JSON o un archivo específico de tipo .avsc. Este archivo describe el nombre, tipo y restricciones de los datos [Kreps *et al.*, 2011]. Avro es independiente del lenguaje de programación, lo que permite la interoperabilidad entre diferentes sistemas y lenguajes. Además, ofrece la flexibilidad de permitir la evolución de los datos a lo largo del tiempo, facilitando así las actualizaciones y cambios en los esquemas de datos [Kreps *et al.*, 2011].

Dentro del ecosistema de Apache Kafka, el Schema Registry desempeña un papel crucial al proporcionar un sistema centralizado para gestionar las versiones de estos esquemas Avro, garantizando que los productores y consumidores estén utilizando versiones compatibles de esquemas.

La relación de Avro con Apache Kafka radica en la necesidad de mantener una coherencia y fiabilidad de los datos a lo largo de las transacciones en un sistema de mensajería distribuido como Kafka. Los mensajes que se intercambian a través de los tópicos de Kafka son, en su

esencia, arrays de bytes. La estructura y el significado de estos bytes dependen totalmente del productor y del consumidor. Para asegurar que el productor y el consumidor tengan una visión común y consistente de la estructura de los datos, se emplean los esquemas [Kleppmann, 2017].

En particular, Avro se utiliza ampliamente con Kafka porque sus esquemas son compactos, rápidos y su evolución permite una compatibilidad hacia atrás y hacia adelante. Con Avro, los productores pueden enviar mensajes codificados con un esquema, mientras los consumidores pueden decodificar estos mensajes con un esquema más antiguo o más nuevo, siempre y cuando sean compatibles [Kleppmann, 2017]. Esto facilita una evolución segura de los esquemas en un sistema en tiempo real, de gran volumen y en constante evolución, como los que están construidos alrededor de Apache Kafka.

### 2.6.5. Kafka Connect

Kafka Connect, parte de la plataforma Apache Kafka, es un marco de trabajo para la integración de sistemas de datos distribuidos en tiempo real. Esta herramienta facilita la importación y exportación de datos hacia y desde Apache Kafka [Kafka, ]. Como se menciona en la documentación oficial, Kafka Connect proporciona un ”framework para construir conectores que conectan Kafka con sistemas externos, permitiendo así la importación y exportación de datos en Kafka”[Kafka, ].

Para entender mejor el funcionamiento interno de Kafka Connect y cómo logra esta integración eficiente, es esencial familiarizarse con sus componentes principales:

- **Conectores:** Son los componentes principales que definen de dónde se extraerán los datos o hacia dónde se enviarán. Estos conectores pueden ser de fuente (importando datos a Kafka) o de sumidero (exportando datos de Kafka a sistemas externos).
- **Tareas (Tasks):** Son las unidades de trabajo que realizan la copia de datos. Cada conector puede dividir el trabajo en múltiples tareas para lograr paralelismo y mejorar la eficiencia.
- **Trabajadores (Workers):** Son procesos que ejecutan conectores y tareas. Pueden ser desplegados en modo independiente o en modo distribuido, lo que permite a Kafka Connect escalar y ser resistente a fallos.
- **Transformaciones (Transforms):** Permiten modificar los registros a medida que se copian entre sistemas. Esto puede incluir cosas como cambiar claves o valores, filtrar registros específicos, etc.
- **Convertidores (Converters):** Son responsables de convertir los datos entre el formato que Kafka utiliza y el formato que el conector requiere.

Continuando, la posibilidad de integrar datos en tiempo real desde diversas fuentes es una de las funcionalidades clave que Kafka Connect proporciona. Mediante sus conectores, los usuarios pueden integrar sistemas de archivos, bases de datos, servicios en la nube y otros sistemas de datos con Kafka, eliminando la necesidad de escribir código personalizado para cada integración.

Los conectores de Kafka Connect están diseñados para ser escalables y tolerantes a fallos, asegurando una transferencia de datos confiable y sin pérdidas, incluso bajo circunstancias adversas. Adicionalmente, Kafka Connect proporciona una API REST para la configuración y el monitoreo de los flujos de datos.

En resumen, Kafka Connect es una herramienta potente y flexible para la integración de datos en tiempo real, que permite la importación y exportación de datos en Apache Kafka [Kafka, ]. Su habilidad para integrar múltiples fuentes de datos y su escalabilidad hacen de Kafka Connect una herramienta esencial para las operaciones de procesamiento de datos en tiempo real.

## 2.7. Cuadros de Mando (Dashboards)

Los cuadros de mando, también conocidos como dashboards, son herramientas de visualización de datos que presentan la información de manera clara, concisa y en tiempo real para facilitar el monitoreo, la medición y la toma de decisiones dentro de una organización [Eckerson, 2010].

Estas herramientas, a menudo impulsadas por tecnologías de inteligencia de negocios (BI) y análisis de datos, están diseñadas para ayudar a las empresas a alcanzar sus objetivos y ejecutar sus planes estratégicos. Proporcionan a los responsables de la toma de decisiones una visión global de los indicadores clave de rendimiento (KPI) de la empresa, así como de otros datos relevantes para el negocio [Few, 2013].

Los Dashboards permiten a los usuarios transformar grandes volúmenes de datos en información comprensible y procesable, destacando tendencias, comparaciones y excepciones [Shneiderman, 1996]. Con esta capacidad de visualización, las organizaciones pueden identificar rápidamente áreas de interés o preocupación, lo que facilita la adaptación a cambios en el entorno empresarial y la toma de decisiones informadas [Eckerson, 2010].

Al presentar información en tiempo real, los dashboards permiten una respuesta rápida a los cambios, proporcionando la oportunidad de ajustar estrategias y operaciones según sea necesario. Además, con el análisis predictivo y la modelización de escenarios, los dashboards también pueden ayudar a prever tendencias futuras, contribuyendo así a la planificación estratégica [Turban *et al.*, 2011].

En resumen, los beneficios clave de los dashboards incluyen el monitoreo en tiempo real, la

toma de decisiones basada en datos, la detección temprana de problemas, la identificación de oportunidades y la previsión de tendencias. Estas características permiten a los agentes de la empresa estar en una mejor posición para tomar decisiones efectivas y oportunas, mejorando así el rendimiento organizacional [Turban *et al.*, 2011].

## 2.8. Monitoreo de Sistemas

El monitoreo de sistemas es esencial en la arquitectura de cualquier solución de tecnología de la información, especialmente en entornos distribuidos. Este proceso implica la recopilación, procesamiento y análisis de datos y métricas de varios sistemas y servicios para asegurar su correcto funcionamiento. Las herramientas de monitoreo proporcionan la capacidad de detectar problemas antes de que afecten a los usuarios finales y ayudan a entender el comportamiento del sistema para optimizar su rendimiento y eficiencia [Louden, 2002].

### 2.8.1. Prometheus

Prometheus es un sistema de monitoreo y alerta de código abierto, diseñado específicamente para manejar el escalado y la confiabilidad en contextos de microservicios y contenedores [Ines y Turnbull, 2018b]. Se adhiere a los principios de la Cloud Native Computing Foundation (CNCF) y se ha convertido en una parte integral de muchas implementaciones de nube nativa.

Prometheus proporciona un modelo de datos multidimensional con series de tiempo, recopilación de métricas basada en el protocolo de extracción (pull), y soporte para consultas precisas con su lenguaje de consulta específico, PromQL. También ofrece capacidades robustas de generación de alertas y tiene un ecosistema activo de bibliotecas y exportadores de terceros, lo que lo hace altamente flexible y adaptable a diferentes escenarios de monitoreo [Ines y Turnbull, 2018b, Authors, 2021].

### 2.8.2. Grafana

Grafana, por otro lado, es una popular herramienta de código abierto para visualizar y analizar datos métricos en tiempo real [Labs, 2021]. Es conocida por su arquitectura modular y extensible que soporta múltiples fuentes de datos y permite integrar varios plugins de visualización.

Además de la visualización de datos en tiempo real, Grafana ofrece una amplia variedad de formatos de visualización, desde gráficos hasta mapas de calor y gráficos de histogramas. Permite personalizar paneles y visualizaciones y proporciona la capacidad de definir alertas.

Su capacidad para integrarse sin problemas con otras herramientas de monitoreo, como Prometheus, InfluxDB y Elasticsearch, hace de Grafana una solución completa para el análisis y monitoreo de datos en tiempo real [Labs, 2021, Ines y Turnbull, 2018a].

# CAPÍTULO 3
# PROPUESTA DE SOLUCIÓN

## 3.1. Arquitectura Propuesta: Componentes y Tecnologías Implementadas

**Consideraciones de Implementación**

Es importante aclarar que GTP, acrónimo de Global Tech Platform, es una plataforma que se encuentra en la red interna de Walmart International (US), mientras que CL se refiere a la red de Walmart Chile. Para simplificar, el diagrama muestra la API publicando datos a los tópicos de Kafka en GTP (ver figura 2), sin embargo, es importante tener en cuenta que, en realidad, los datos se publican en un tópico en CL y se replican en US.

Además, es crucial asegurarse de que todas las herramientas y tecnologías utilizadas estén dentro del conjunto de soluciones permitidas por Walmart International.

**Recursos y Tecnologías**

- **Google Cloud Platform (GCP)**: Esta robusta plataforma de servicios de computación en la nube brindará el ambiente necesario para el despliegue y funcionamiento de la solución. Dentro de GCP, se hará uso específico de:
    - **BigQuery**: Permitirá almacenar y analizar grandes conjuntos de datos.
    - **DataStudio**: Se utilizará para la creación de informes y visualizaciones de datos interactivos.
- **Apache Kafka**: Este será el sistema de mensajería en tiempo real, utilizado para la transmisión de datos entre las aplicaciones y servicios.
- **Kafka Connect**: Este componente de Kafka permitirá crear conectores reutilizables entre Kafka y otros sistemas.
- **Tableau**: Esta plataforma de visualización de datos ayudará a interpretar los datos de manera efectiva y eficiente.

El diseño tecnológico del proyecto se basa en la selección de herramientas que ofrecen un equilibrio entre funcionalidad, costos y facilidad de integración. La elección de BigQuery como solución de almacenamiento de datos se realizó tras considerar varias opciones dentro de las tecnologías disponibles de Walmart International. BigQuery sobresalió por ser la única opción de Big Data As A Service, dado que Azure Synapse no está disponible y Azure SQL, aunque accesible, no está optimizado como opción para Data Warehousing, lo que podría resultar en mayores costos a largo plazo.

Optar por otras soluciones habría implicado el uso de métodos como API proxy para conectarse a la red interna, lo que podría desencadenar en una mayor utilización de recursos y problemas futuros, especialmente considerando la posibilidad de que la información sensible no se utilizara en adelante.

Es relevante señalar que los datos expuestos en la oferta no están normalizados. Con BigQuery, es posible aprovechar esta característica para potenciar el rendimiento.

En cuanto a la transmisión de datos, Kafka y Kafka Connect son las herramientas escogidas. La API de Promise Engine, Lastmile, se replica en múltiples ocasiones debido a su naturaleza de sistema distribuido. Así, los datos de oferta, reserva y confirmación se publican en tópicos de Kafka. Kafka Connect sirve como una herramienta efectiva para la transmisión de datos entre Kafka y BigQuery, ofreciendo una solución como servicio que se adapta sin inconvenientes a la plataforma de Walmart.

Finalmente, para la visualización de datos, se optó por Tableau. La elección se orientó hacia un software enfocado en la inteligencia empresarial que posibilitara la interacción con los datos. PowerBI Desktop fue descartado por su incompatibilidad con Mac. Data Studio también fue ponderado, dada su naturaleza de herramienta basada en la web.

### 3.1.1. Estado Previo a la Implementación

Antes de la propuesta de dicha solución, el Promise Engine gestionaba tres eventos; sin embargo, únicamente la data de la oferta de ventanas se almacenaba en Dremio, que forma parte de la arquitectura de CL. Con la solución propuesta, el Promise Engine no solo será capaz de transmitir a Kafka los eventos de oferta, reserva y confirmación, sino que, de manera crucial, también resultará compatible con la tecnología de GTP.

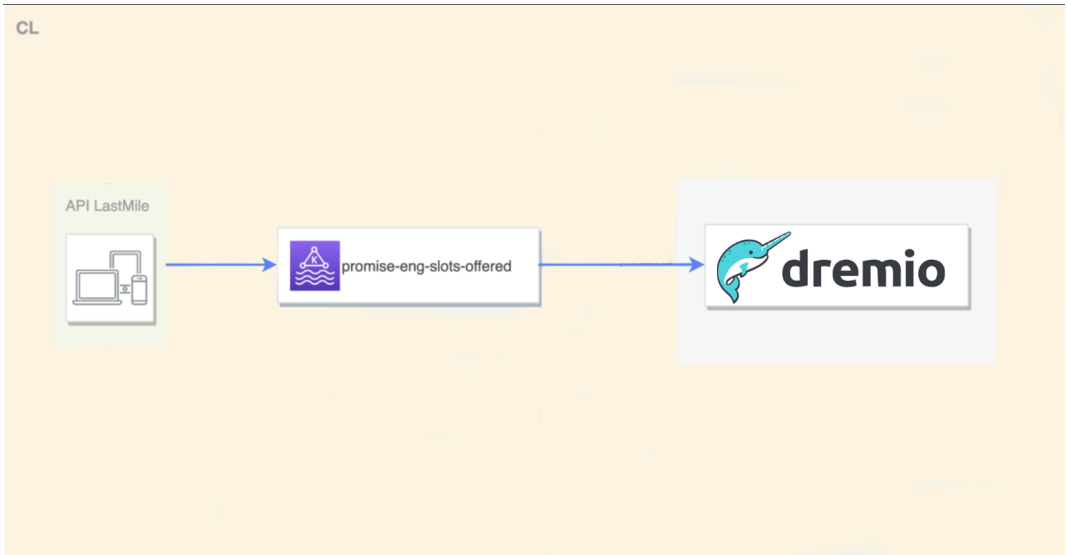



Figura 2: Arquitectura de datos inicial de Promise Engine.
Fuente: Elaboración Propia.

### 3.1.2. Proceso de migración

Considerando la migración de Walmart CL a la plataforma internacional, es esencial coordinar dicha migración con la implementación de este sistema. Por ello, se propone replicar los tópicos de la arquitectura ubicada en Chile a los correspondientes tópicos en GTP.

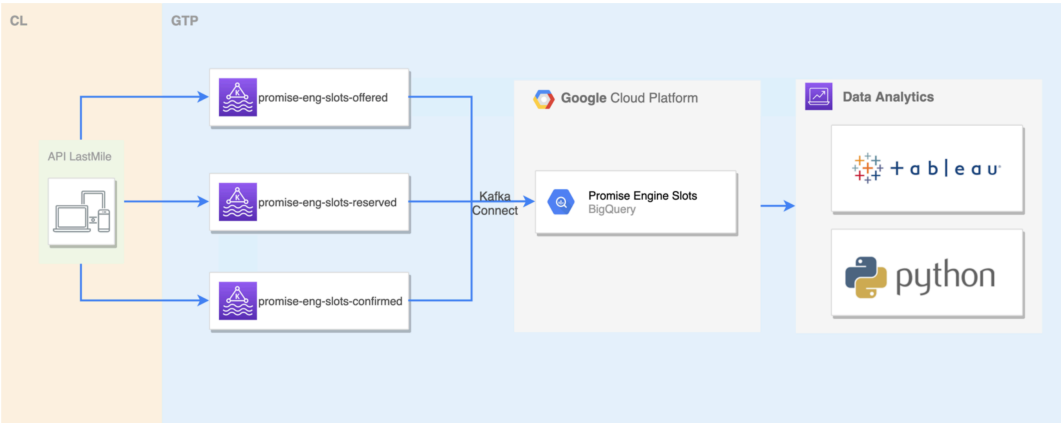



Figura 3: Arquitectura de datos propuesta para la migración de Promise Engine.
Fuente: Elaboración Propia.

### 3.1.3. Arquitectura Final

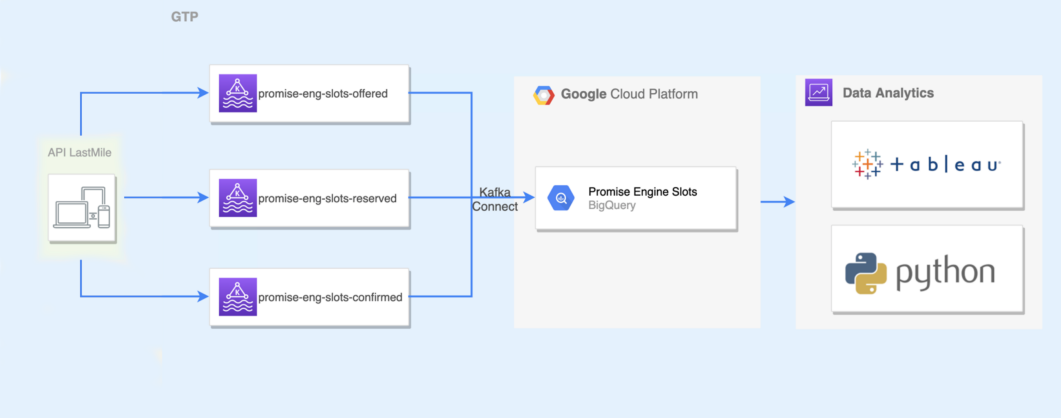



Figura 4: Arquitectura de datos final de Promise Engine.
Fuente: Elaboración Propia.

Una vez que se alcance la fase final de la migración, se espera que todo el sistema esté integrado en GTP. La solución estará completamente alineada con la migración del artefacto. Solo será necesario reconfigurar los tópicos, pero esta vez dentro del ecosistema de GTP.

## 3.2. Descripción de los Datos

Como se indicó anteriormente, la API Promise Engine, Lastmile, tiene la responsabilidad de exponer de forma continua tres eventos mediante la mensajería de Kafka. Estos mensajes deben estructurarse correctamente antes de su envío, lo cual subraya la importancia de definir su esquema de manera precisa al planear una solución que utilice Apache Kafka.

A continuación, se presenta una tabla que detalla cada uno de los campos correspondientes al evento de oferta. Es importante señalar que algunos de estos campos también están presentes en los eventos de reserva y confirmación, dado que estos eventos comparten un esquema similar. Esto garantiza consistencia y uniformidad en el flujo de la información a lo largo de los distintos eventos.

Tabla 1: Datos de la oferta de ventana.
Fuente: Elaboración Propia.

| *Nombre del Schema de la Base de Datos* | *Tipo de dato de la columna/campo* | *Definición desde el negocio/bd del uso del dato* |
|---|---|---|
| **requestTime** | TIMESTAMP | Timestamp en el que se hizo la request |
| **shippingGroup** | STRING | ShippingGroup Id |
| **action** | STRING | Acción a realizar, en este caso Oferta |
| **channel** | STRING | Canal de Venta |
| **capability** | STRING | Metodo envió, Pickup ó HomeDelivery que el cliente selecciono para consultar |
| **deliveryType** | STRING | Tipo de Delivery |
| **regionId** | INTEGER | Id de la región |
| **communeId** | INTEGER | Id de la Comuna |
| **pickingStoreId** | INTEGER | Id de la store de la cual se realiza el picking |
| **storeId** | INTEGER | Id de la store donde se realiza la compra |
| **slotDate** | DATE | Fecha del slot (Despacho al usuario) |
| **slotStartTime** | STRING | Inicio de rango hora para entrega o retiro de la compra |
| **slotEndTime** | STRING | Finalización de rango hora para entrega o retiro de la compra |
| **slotCost** | INTEGER | Costo del shipping del slot |
| **shiftDate** | DATE | Fecha de Picking |
| **shiftStartTime** | STRING | Inicio del rango hora para reservar la capacidad de picking |
| **shiftId** | INTEGER | Id del slot de picking |
| **shiftEndTime** | STRING | Finalización del rango hora para reservar la capacidad de picking |
| **size** | INTEGER | Tamaño de la orden |
| **inventoryType** | STRING | Tipo de inventario, puede ser Centralized u OnHand |
| **vendorNumber** | STRING | Numero del vendedor |
| **units** | INTEGER | Unidades de la compra |

### 3.2.1. Esquema

El esquema de cada mensaje desempeñó una función crucial para garantizar que los mensajes enviados por el artefacto de Lastmile, implementado en JavaScript, se formatearan correctamente. En este caso específico, no fue necesario emplear el Schema Registry en Kafka Connect, ya que los mensajes son bastante sencillos y es suficiente con definir todos los campos con los tipos correspondientes en BigQuery. Sin embargo, para mensajes más complejos, la utilización de Schema Registry se vuelve altamente recomendable.

A continuación, se muestra el esquema de los eventos de reserva. Es importante destacar que el esquema para los eventos de confirmación es idéntico.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "Promise Engine Reserve Event",
  "description": "An event triggered by the reservation of PE",
  "type": "object",
  "properties": {
    "requestTime": {
      "type": "timestamp"
    },
    "shippingGroup": {
      "type": "string"
    },
    "shiftId": {
      "type": "integer"
    },
```

```
    "slotId": {
      "type": "integer"
    },
    "channel": {
      "type": "string"
    },
    "deliveryType": {
      "type": "string"
    },
    "dropshipDate": {
      "type": "string"
    },
    "units": {
      "type": "integer"
    }
  },
  "required": [
    "requestTime"
  ]
}
```

Listing 1: Ejemplo del Schema de la reserva

## 3.3. Establecimiento de Hipótesis

Dentro del contexto de esta investigación, se formularon una serie de hipótesis destinadas a evaluar características, funcionalidades y aspectos específicos de la propuesta. Cada hipótesis juega un papel fundamental en el proceso de verificación, ofreciendo una oportunidad valiosa para medir la eficacia de la solución propuesta. Estas hipótesis guiarán los experimentos y pruebas que se llevarán a cabo y servirán como base para posibles ajustes y optimizaciones antes de la implementación definitiva de la solución.

Las hipótesis que se abordarán son las siguientes:

Tabla 2: Hipótesis a estudiar.
Fuente: Elaboración Propia.

| **Hipótesis** | **Categoría** |
|---|---|
| El costo de entrega influye en la decisión de compra. | Costo de Slot |
| El costo de entrega influye en el tipo de entrega elegido por el cliente (s2s). | Costo de Slot |
| El cliente tiende a elegir la primera ranura de entrega disponible. (HD) | Elección de Slot |
| El cliente tiende a elegir HD cuando el tamaño del producto es grande. | Tamaño del Producto |

Es importante señalar que también se espera examinar el estado de las ventanas en los

distintos segmentos geográficos:

- Zona Norte Grande
- Zona Norte Chico
- Zona Central de Chile
- Región Metropolitana (RM)
- Zona Sur
- Zona Austral (En caso de ser posible)

Esto garantiza que la solución propuesta sea eficaz y cumpla con las necesidades y expectativas de los usuarios en diversas regiones. El objetivo trasciende la viabilidad técnica de la solución; se busca garantizar que la propuesta responde de manera efectiva a las necesidades y expectativas de los usuarios. Para simplificar el proceso de validación de estas hipótesis, se han agrupado por categorías.

Además, se establecen como deseables los siguientes Indicadores Clave de Rendimiento (KPIs) y reportes adicionales para proporcionar una comprensión más detallada del rendimiento de la solución propuesta:

- Proporción de Conversión del flujo de venta:
  - Oferta $\rightarrow$ Reserva
  - Reserva $\rightarrow$ Confirmación
- Segmentación por Tipo de Inventario
- Proporción de ofertas de ventana que sean:
  - Entrega el mismo día
  - Picking el día siguiente

La meta es proporcionar una visión comprensiva que facilite una implementación exitosa y adaptada a las necesidades de la situación.

## 3.4. Historias de usuario y épicas

Walmart CL adopta una metodología ágil similar a Scrumban. En línea con este enfoque, se desarrollaron spikes y se definieron historias de usuario.

### 3.4.1. Spikes

Antes de establecer las Historias de Usuario, es esencial llevar a cabo un análisis exhaustivo de los ”Spikes”. Estos términos, comunes en el desarrollo ágil de software, hacen referencia a tareas de investigación diseñadas para solucionar incertidumbres técnicas o conceptuales que podrían representar obstáculos o riesgos para el proyecto.

Los Spikes son esenciales para comprender a fondo los desafíos técnicos o los problemas complejos que podrían surgir al implementar las características descritas en las historias de usuario. Además, contribuyen a evaluar la viabilidad de las soluciones sugeridas y a estimar con más precisión el tiempo y los recursos necesarios para desarrollar una funcionalidad determinada.

En la fase inicial del proyecto, se llevó a cabo un meticuloso proceso de identificación y análisis de Spikes para las funcionalidades clave sugeridas. Esto incluyó diversas actividades de investigación, como la revisión de documentación técnica, pruebas de concepto, sesiones de brainstorming y conversaciones detalladas con el equipo de desarrollo. Estos elementos permitirán prever y manejar los posibles desafíos técnicos y conceptuales durante la implementación de las historias de usuario establecidas para el sistema.

Tabla 3: Spike-01.
Fuente: Elaboración Propia.

| | |
|---|---|
| **ID** | SPK-01 |
| **Nombre** | Discovery planteamiento PE |
| **Contexto** | Actualmente no se cuenta con un sistema para el análisis y seguimiento de los datos, lo anterior provoca que la toma de decisiones sea menos informada, guiada más por el feeling humano que implementando una estrategia de decisiones basada en datos.<br>Por lo tanto se debe definir la arquitectura correcta para empezar el desarrollo de esta y la estrategia de datos que se tendrá. |
| **Entregable** | Set de preguntas/hipótesis del Promise Engine que pudieran ser respondidas con la data expuesta actualmente del producto<br>Set de hipótesis desde la mirada holística del producto a largo plazo, que escapan del procesamiento de la data actual. Como por ej: múltiples oportunidades de envío a cliente ubicado en la coordenada (x,y) |

Tabla 4: Spike-02.
Fuente: Elaboración Propia.

| | |
|---|---|
| **ID** | SPK-02 |
| **Nombre** | Diseño de modelo y arquitectura de datos |
| **Contexto** | Actualmente no se cuenta con un sistema para el análisis y seguimiento de los datos, lo anterior provoca que la toma de decisiones sea menos informada, guiada más por el feeling humano que implementando una estrategia de decisiones basada en datos. Por lo tanto se debe definir la arquitectura correcta para empezar el desarrollo de esta y la estrategia de datos que se tendrá. |
| **Entregable** | Este discovery debe ser apoyado por el equipo de exploración de datos (Felipe Piña) y guiado por la estrategia de data de la compañía (Sebastián Santiago). Existe un equipo externo ”Spike”que podrá realizar acompañamiento en la identificación de modelos de datos para las hipótesis. |

### 3.4.2. Historias de Usuario

A continuación, se detallan algunas de las historias de usuario más relevantes que se identificaron al momento de querer iniciar el desarrollo. Estas historias brindan una visión general de los objetivos y funcionalidades clave que se espera implementar en el sistema. Para iniciar el desarrollo de este proyecto, fue esencial comprender y establecer los requisitos y necesidades de los usuarios finales del sistema propuesto. Una de las formas más efectivas de lograrlo es a través de la definición de las ’Historias de Usuario’, una técnica de ingeniería de software que permite describir las características y funcionalidades del sistema desde la perspectiva del usuario final, teniendo en cuenta las necesidades, objetivos y la manera en que interactuarán con el sistema.

Estas historias de usuario resultantes son un componente vital en el proceso de desarrollo de software, proporcionando una guía clara y detallada de los requisitos y permitiendo priorizar y planificar eficientemente las etapas de desarrollo.

Por lo tanto, en esta sección, se presentaran algunas de las historias de usuario más relevantes que se identificaron durante la etapa inicial del proyecto. Cada una de ellas proporciona una descripción detallada de una funcionalidad clave del sistema, incluyendo el papel del usuario, la acción que desea realizar y el beneficio o resultado esperado. Este conjunto de historias de usuario proporcionará una visión integral sólida de los objetivos y funcionalidades clave que se espera implementar en el sistema, facilitando el proceso de diseño e implementación de la solución sugerida.

Tabla 5: Historia de Usuario-01.
Fuente: Elaboración Propia.

| | |
|---|---|
| **ID** | HDU-01 |
| **Nombre** | HDU1: Implementación diseño de arquitectura |
| **Descripción** | COMO: equipo de kronos QUIERO: tener disponible herramientas PARA: hacer análisis de los datos de los clientes y obtener insights de estos. |
| **Antecedentes** | ▪ Spike de diseño de arquitectura:<br>▪ Información de la oferta, reserva y confirmación actualmente se está exponiendo en un topic.<br>▪ La oferta se viene registrando desde finales del 2019, sin embargo hay columnas que son más nuevas, por lo tanto no hay registros más antiguos que 2020 o 2021 (Nota: Por confirmar las columnas que son más nuevas, aún se está explorando la data).<br>▪ En dremio se tiene lastmile-slots-offered-sink, en donde se puede ver los datos expuestos de la oferta.<br>▪ Actualmente se tiene la base de datos de MongoDB, que si bien no guarda toda la información se puede utilizar para tener datos más completos de la reserva y confirmación previo al 2021.<br>▪ Investigación realizada sobre las opciones disponibles en GTP, a partir de esta se llegó al acuerdo de utilizar GCP.<br>▪ El equipo de Project Blackwell (Equipo de Data Science que ya está en Walmart Int.) realizó un mapeo de infraestructura (Infrastructure solution mapping), después de una breve explicación Thomas Lauritzen también plantea GCP como la mejor opción para el proyecto dado que aún no se encuentra disponible la alternativa de azure y otras alternativas representan mucho problema con infosec. |
| **Criterios de Aceptación** | ▪ DADO: la necesidad de almacenar mi data CUANDO: ejecute una query con data para testing ENTONCES: obtengo resultados<br>▪ DADO: las especificaciones de la arquitectura diseñada CUANDO: se realizan las especificaciones pedidasENTONCES: la infraestructura entregada debe coincidir con la diseñada |

Tabla 6: Historia de Usuario-02.
Fuente: Elaboración Propia.

| | |
|---|---|
| **ID** | HDU-02 |
| **Nombre** | Procesamiento de Datos |
| **Descripción** | COMO: equipo Kronos QUIERO: trabajar en modelos de datos PARA: obtener insights de los clientes |
| **Antecedentes** | ▪ La arquitectura diseñada<br>▪ Dado que se está trabajando con data histórica es necesario consultar en Dremio y MongoDB. Esto significa que es probable que se deba implementar algo adicional para pasar la data que está en CL a internacional. |
| **Criterios de Aceptación** | ▪ DADO: la infraestructura de datos CUANDO: se requiera hacer el levantamiento inicial de la data ENTONCES: se debe contar con las herramientas necesarias para habilitar la historia de observabilidad.<br>▪ DADO: una construcción de querys CUANDO: se requiera obtener información respecto a qué representa cada una ENTONCES: se debe contar con la documentación respectiva<br>▪ DADO: el monitoreo operacional CUANDO: se requiera obtener información respecto al performance de los artefactos ENTONCES: se debe contar con un dashboard de visibilidad |

Tabla 7: Historia de Usuario-03.
Fuente: Elaboración Propia.

| | |
|---|---|
| **ID** | HDU-03 |
| **Nombre** | HDU3: Observabilidad Data Promise Engine |
| **Descripción** | COMO: equipo Kronos QUIERO: tener un dashboard de visibilidad de la subasta PARA: generar los insights apropiados de cara a las hipótesis de producto |
| **Antecedentes** | El output de las historias anteriores |
| **Criterios de Aceptación** | ▪ DADO: la necesidad de monitorear la data del promise engine CUANDO: se requiera accionar o gestionar esta ENTONCES: se debe contar con dashboard de visibilidad<br>▪ DADO: una construcción de dashboards CUANDO: se requiera obtener información respecto a qué representa cada uno ENTONCES: se debe contar con la documentación respectiva |

Tabla 8: Historias de Usuario 04|05|06.
Fuente: Elaboración Propia.

| | |
|---|---|
| **ID** | HDU-04 \| HDU-05 \| HDU-06 (Al ser 3 categorias de hipotesis se crearon 3 historias) |
| **Nombre** | HDU: Creación de reportes para hipótesis |
| **Descripción** | COMO: equipo Kronos QUIERO: tener los informes respecto a cada hipótesis PARA: poder tomar decisiones estratégicas acerca de la evolución del producto (respaldadas con datos) |
| **Antecedentes** | Output de historias anteriores |
| **Criterios de Aceptación** | DADO: El informe de cada hipótesis (Nota: informe sería el conjunto de dashboards y anotaciones en estos) CUANDO: busca los resultados correspondientes a estas ENTONCES: se debe ver el listado de reportes de cada métrica y la conclusión de la hipótesis (c/u) |

Existen tres categorías de hipótesis relacionadas con la elección por parte del cliente de la ventana de despacho, el costo del slot y el tamaño de los productos. Por lo que en las últimas historias (ver tabla 8), cada una corresponde a una de estas categorías.

Es importante señalar que, para el desarrollo de la investigación, únicamente se consideraron las ventanas presentes en el catálogo extendido. A pesar de que la data para supermercados está disponible, estos carecen de un identificador ”shippingGroupId”. Dado que dicho identificador aparece como NULL, esto impide la validación de las hipótesis propuestas.

## 3.5. Exposición de data

El artefacto de lastmile del sistema actualmente emplea la biblioteca KafkaJS para exponer los eventos fundamentales. No obstante, se ha determinado que es necesario llevar a cabo una reestructuración de su implementación. A pesar de que esta actividad no figura en las historias de usuario previamente enumeradas, durante el transcurso del proyecto, fue necesario efectuar una migración a Kafka 2.4. En el proceso de dicha migración, se aprovechó la oportunidad para realizar modificaciones en el cliente de Kafka.

```
1
2  const kafkaBrokers = [KAFKA_BROKER_24_1, KAFKA_BROKER_24_2, KAFKA_BROKER_24_3, .K
3
4  let kafkaInstance
5  let kafkaConsumer
6  let kafkaProducer
7
8  function initializeKafkaClient () {
9    if (kafkaInstance === undefined || kafkaInstance == null) {
10     kafkaInstance = new Kafka({
11       brokers: kafkaBrokers,
12       clientId: `lastmile-api-${process.env.NODE_ENV}-group`,
13       requestTimeout: 5 * 60 * 1000,
14       ssl: kafkaUtils.getSslConfig(),
15       retry: {
16         retries: 10,
17         maxInFlightRequests: 1
18       },
19       logLevel: logLevel.INFO
20     })
21   }
22 }
23
24 async function initializeKafkaConsumer () {
25   try {
26     initializeKafkaClient()
27     if (kafkaConsumer === undefined || kafkaConsumer == null) {
28       kafkaConsumer = kafkaInstance.consumer({
29         groupId: process.env.KAFKA_GROUP_ID,
30         sessionTimeout: 5 * 60 * 1000,
31         heartbeatInterval: 45 * 1000,
32         maxInFlightRequests: process.env.KAFKA_PRODUCER_INFLIGHT_REQUEST_SUB
33       })
34       await kafkaConsumer.connect()
35     }
36   } catch (error) {
37     log.error('Error initializating kafka consumer', { error: error.message, stac
```

```
  }
}

async function initializeKafkaProducer () {
  try {
    initializeKafkaClient()
    if (kafkaProducer === undefined || kafkaProducer == null) {
      kafkaProducer = kafkaInstance.producer({
        createPartitioner: Partitioners.DefaultPartitioner,
        maxInFlightRequests: process.env.KAFKA_PRODUCER_INFLIGHT_REQUEST_PUB
      })
      await kafkaProducer.connect()
    }
  } catch (error) {
    log.error('Error initializating kafka producer', { error: error.message, stac
  }
}
```

Listing 2: Parte del código del cliente de kafka en el artefacto de Promise Engine

De los cambios más importantes que se implementaron, se destacan:

- **Cambio en creación de Producers:** Los productores en Kafka son los encargados de enviar los mensajes a los tópicos. En la nueva implementación, se optó por declarar un único productor (kafkaProducer) que se encargará de publicar los mensajes en los diferentes tópicos de Kafka. Anteriormente, cada vez que era necesario publicar un mensaje, se creaba un nuevo productor. Ese enfoque resultaba ineficiente y consumía más recursos de los necesarios. Ahora, con un solo productor que puede ser reutilizado para publicar en diferentes tópicos (ver función en *initializeKafkaProducer*), se mejora el rendimiento y se optimiza el uso de recursos.
- **Número de Consumers:** Los consumidores son los encargados de recibir los mensajes de los tópicos en Kafka. Antes, se creaba un nuevo consumidor por cada tópico, lo cual también resultaba ineficiente. En la actualización, se optó por implementar un único consumidor (kafkaConsumer) que puede suscribirse a varios tópicos (ver función en *initializeKafkaConsumer*). Esta práctica es recomendada para mejorar el rendimiento, ya que minimiza la latencia y la sobrecarga de la creación y destrucción de consumidores. Al mantener un único consumidor por instancia, también se facilita el seguimiento y la gestión de los consumidores, además de mejorar la resiliencia del sistema al evitar la dependencia de múltiples consumidores.

En la implementación original del artefacto de lastmile, la definición de los productores y consumidores se encontraba dentro de la misma función de consumo. Sin embargo, con la actualización, se optó por declararlos como variables separadas. Este enfoque evita la

inicialización de múltiples productores y asegura que cada tópico tenga un único consumidor compartido, siguiendo las recomendaciones sugeridas.

### 3.5.1. Creación de tablas en BigQuery

La creación de tablas es un paso fundamental en la preparación y organización de los datos para su posterior análisis y consulta en BigQuery.

En BigQuery, las tablas se definen utilizando el comando CREATE TABLE, que permite especificar los campos, tipos de datos y otras propiedades relevantes. El esquema de una tabla en BigQuery proporciona una descripción detallada de las columnas y las características, como el tipo de datos, la longitud, la precisión y la opción de ser nulos o requeridos.

La correcta configuración del esquema de cada tabla es crucial, tal como se discutió anteriormente. Es fundamental considerar con detenimiento los tipos de datos y las restricciones necesarias para asegurar la integridad y la calidad de los datos almacenados. En el contexto específico de este proyecto, es esencial que los tipos de datos de las tablas sean coherentes con los que se envían en los mensajes de Kafka.

Al garantizar que los tipos de datos en el esquema de las tablas coincidan con los tipos de datos en los mensajes de Kafka, se evitan posibles problemas de integridad y se facilita el proceso de carga y consulta de datos. Esto permite un flujo de datos consistente y confiable a lo largo de todo el sistema.

```
 "CREATE TABLE wmt-gcp-promise-engine-prod.gtp_promise_engine_data.cl-lastmile-pr
(
  requestTime TIMESTAMP NOT NULL OPTIONS(description="Timestamp del request"),
  shippingGroup STRING,
  action STRING,
  channel STRING,
  capability STRING,
  deliveryType STRING,
  regionId INT64,
  communeId INT64,
  communeName STRING,
  pickingStoreId INT64,
  pickingStoreName STRING,
  storeId INT64,
  storeName STRING,
  slotDate DATE,
  slotStartTime STRING,
  slotEndTime STRING,
  slotCost INT64,
  shiftDate DATE,
```

```
  shiftStartTime STRING,
  shiftId INT64,
  shiftEndTime STRING,
  size INT64,
  inventoryType STRING,
  vendorNumber STRING,
  units INT64
)
PARTITION BY TIMESTAMP_TRUNC(_PARTITIONTIME, MONTH)
OPTIONS(
  require_partition_filter=true
);"
```

Listing 3: Query para la creación de tabla de oferta

```
"CREATE TABLE wmt-gcp-promise-engine-prod.gtp_promise_engine_data.cl-lastmile-pro
(
  requestTime TIMESTAMP OPTIONS(description="when request was sent"),
  shippingGroup STRING OPTIONS(description="shipping group id"),
  shiftId INT64 OPTIONS(description="picking slot identifier"),
  slotId INT64 OPTIONS(description="delivery slot identifier"),
  channel STRING OPTIONS(description="sales channel (gm/gr)"),
  deliveryType STRING OPTIONS(description="delivery type (home delivery/pick-up)"
  dropshipDate STRING OPTIONS(description="vendor delivery date"),
  units INT64 OPTIONS(description="total units for shipping group")
)
PARTITION BY TIMESTAMP_TRUNC(_PARTITIONTIME, MONTH)
OPTIONS(
  require_partition_filter=true
);"
```

Listing 4: Query para la creación de tabla de reserva

Se debe destacar que *la tabla de confirmación es igual que la de reserva*.

Todas las tablas en BigQuery se encuentran particionadas por mes. Esta decisión se basa en una serie de ventajas significativas en términos de rendimiento y costos.

La partición por mes proporciona mejoras en la eficiencia de las consultas. Al dividir los datos en particiones mensuales, BigQuery tiene la capacidad de omitir automáticamente las particiones que no son relevantes para una consulta específica. Como resultado, reduce el tiempo de respuesta de las consultas, ya que evita el escaneo de datos innecesarios.

Además, la partición mensual simplifica la administración de datos. Permite cargar y eliminar datos en particiones específicas, facilitando las tareas de mantenimiento y gestión a largo plazo. Por ejemplo, es posible cargar nuevos datos mensuales en la partición correspondiente sin afectar el resto de la tabla, optimizando así el proceso de actualización de datos.

Otro beneficio importante se relaciona con el ahorro de costos. Al utilizar el particionamiento por mes, es posible aprovechar las funciones de BigQuery para optimizar los costos de almacenamiento y consultas. BigQuery almacena y factura de forma separada las particiones no utilizadas en una consulta, lo que brinda un mayor control y permite optimizar los costos de almacenamiento y procesamiento de datos.

En resumen, el particionamiento por mes en las tablas de BigQuery ofrece mejoras significativas en el rendimiento de las consultas, facilita la administración de los datos y permite una mayor eficiencia en los costos. Esta estrategia es particularmente valiosa cuando se trabaja con grandes volúmenes de datos y se busca maximizar la eficacia y la rentabilidad en el análisis de datos en BigQuery.

### 3.5.2. Extracción Data Histórica


Figura 5: Extracción de data histórica de Oferta desde Dremio
Fuente: Elaboración Propia.

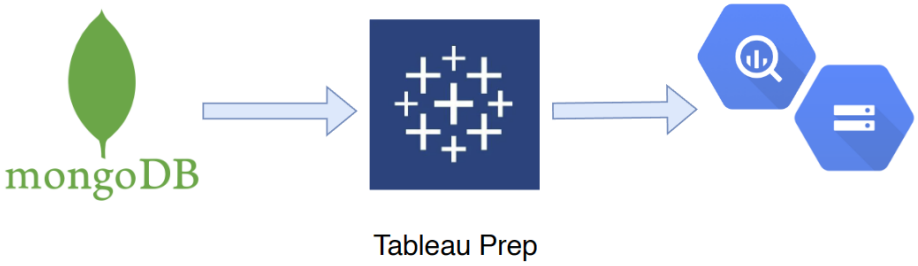



Figura 6: Extracción de data histórica de Reserva y Confirmación desde MongoDB
Fuente: Elaboración Propia.

Dado que los datos deben ser manipulados para que sean compatibles con el esquema de los mensajes enviados a Kafka y, en consecuencia, con las tablas de BigQuery, se optó por implementar un proceso ETL (Extract, Transform, Load) utilizando Tableau Prep (ver Figura 6). Esta elección permitió llevar a cabo la transformación de los datos en un único flujo de trabajo y completar los campos faltantes utilizando la información migrada de la oferta previa.

En este caso particular, los eventos disponibles se limitaban al catálogo extendido y no incluían datos de supermercado. Por lo tanto, se procedió de la siguiente manera:

- Se recuperó la información de *çhannel*" y *"deliveryType*" a partir de los datos históricos de la oferta mediante el uso de SG (match). Esto implicó utilizar técnicas de coincidencia para relacionar los datos existentes con los atributos correspondientes de *çhannel*" y *"deliveryType*".
- Se realizó una transformación de los campos *"shiftId*" y *"slotId*" de STRING a INTEGER. Esto se llevó a cabo para asegurar que los datos estuvieran en el formato adecuado y optimizado para su posterior análisis y consulta.

Al implementar el proceso ETL con Tableau Prep y realizar estas transformaciones específicas, se logró adaptar la data de manera adecuada, garantizando su compatibilidad con el esquema de mensajes de Kafka y las tablas de BigQuery.

Por otro lado, con el fin de extraer los datos desde Dremio, se implementó un extractor de datos sencillo en Python (ver Figura 5). Este extractor tuvo la función de guardar los datos en formato CSV. Dado que estos datos ya contaban con el esquema oficial requerido, no fue necesario realizar ninguna transformación adicional en ellos.

### 3.5.3. Optimización de Dashboards mediante Vistas en BigQuery

Un elemento esencial en la estrategia de gestión y visualización de datos ha sido la implementación de vistas en BigQuery destinadas a la creación de dashboards y reportes. Las vistas en BigQuery funcionan como tablas virtuales, constituidas a partir de consultas SQL predefinidas, lo que permite que no almacenen datos per se, sino que actúen como una ventana referencial a la consulta subyacente.


Figura 7: Vistas en el proyecto de BigQuery
Fuente: Elaboración Propia.

## 3.6. Configuración de Tópicos de Eventos

Para configurar adecuadamente los tópicos de eventos, es esencial definir ciertos parámetros, tales como las *particiones*, *horas de retención* (también conocido como *retention hours*), *retention bytes* y el *tipo de compresión*.

El número de particiones se determina por la cantidad de consumidores que se desean utilizar para leer los datos del tópico. En este caso, dado que se cuenta con 3 pods para Kafka Connect, se tendrán tres consumidores y, por ende, se necesitarán tres particiones.

Las *Horas de Retención* se establecen en 72 horas (3 días). Esto brinda la capacidad de acceder a datos en los tópicos incluso después de los fines de semana.

El cálculo de los *retention bytes* se basa en la fórmula proporcionada por el equipo internacional de Kafka en Walmart:

$$\frac{(\text{Número de Mensajes al día}) \times (\text{Tamaño promedio en bytes}) \times (\text{Días de retención})}{\text{Número de particiones}}$$

Es importante mencionar que en Walmart se utiliza, como estándar, el algoritmo de compresión lz4 en el servidor Kafka. Esto comprime los datos al guardarlos en el disco, resultando en un tamaño real menor al inicialmente calculado.

A continuación, se describen las configuraciones específicas para tres tópicos diferentes: cl-lastmile-promise-engine-slots-offered, cl-lastmile-promise-engine-slots-reserved y cl-lastmile-promise-engine-slots-confirmed.

### 3.6.1. Configuración para `cl-lastmile-promise-engine-slots-offered`

- **partitions**: 3
- **retention.hours**: 72hrs
- **retention.bytes**: Basado en la fórmula, y considerando un promedio diario de 12000000 mensajes de 4750 Bytes cada uno, se obtiene un total de 57000000000 Bytes (equivalente a 57Gb).
- **compression.type**: lz4

### 3.6.2. Configuración para `cl-lastmile-promise-engine-slots-reserved`

- **partitions**: 3
- **retention.hours**: 72hrs
- **retention.bytes**: Usando la misma fórmula, el resultado es aproximadamente 28000000000 Bytes.
- **compression.type**: lz4

### 3.6.3. Configuración para `cl-lastmile-promise-engine-slots-confirmed`

- **partitions**: 3
- **retention.hours**: 72hrs

- **retention.bytes**: Utilizando la fórmula proporcionada, se obtiene un total de aproximadamente 11000000000 Bytes.
- **compression.type**: lz4

## 3.7. Data Pipeline: Kafka Connect

Kafka Connect proporciona una plataforma sólida y escalable para la integración de datos de fuentes externas hacia y desde Apache Kafka. Su configuración es esencial para optimizar el rendimiento y la eficiencia de este proceso. En esta sección, se presentan las configuraciones del ”worker” de los conectores implementados en este proyecto, con una breve justificación de cada una.

### 3.7.1. Configuración del Worker

La configuración del ”worker”se refiere a los parámetros generales que aplican para todas las operaciones del Kafka Connect. Las configuraciones del ”worker”son las siguientes:

```
worker:
    key.converter: org.apache.kafka.connect.storage.StringConverter
    value.converter: org.apache.kafka.connect.json.JsonConverter
    key.converter.schemas.enable: false
    value.converter.schemas.enable: false
    group.id: kronos-kc-ingestor
    security.protocol: SSL
    config.storage.topic: kronos-bq-sink-config
    offset.storage.topic: kronos-bq-sink-offset
    status.storage.topic: kronos-bq-sink-status
    offset.flush.interval.ms: 10000
```

Listing 5: Configuraciones del Worker

- **key.converter**: Se establece en `org.apache.kafka.connect.storage.StringConverter`, lo que significa que las claves se convierten a cadenas de texto.
- **value.converter**: Se establece en `org.apache.kafka.connect.json.JsonConverter`, lo que significa que los valores se convierten a formato JSON.
- **key.converter.schemas.enable** y **value.converter.schemas.enable**: Ambos se establecen en `false`, ya que no se necesita almacenar o recuperar información del esquema en este proyecto.

- **group.id**: Se establece en `kronos-kc-ingestor`, lo que identifica al grupo de consumidores al que pertenece este worker.
- **security.protocol**: Se establece en `SSL`, para la seguridad de la comunicación.
- **config.storage.topic**, **offset.storage.topic**, **status.storage.topic**: Se configuran para almacenar la configuración, el offset y el estado en Kafka respectivamente.
- **offset.flush.interval.ms**: Se establece en 10000ms para controlar la frecuencia con la que se actualizan los offsets en Kafka.

### 3.7.2. Configuración de los Conectores

La configuración de los conectores se refiere a los parámetros que aplican a un conector en particular. Para todos los conectores se utilizaron las mismas configuraciones.

```
connectors:
 - name: bq-connector-event-slots
   config:
        connector.class: com.wepay.kafka.connect.bigquery.BigQuerySinkConnector #
        max.batch.size: 100
        max.retries: 10
        tasks.max: 5
        topics: cl-lastmile-promise-engine-slots-reserved # GTP
        autoCreateTables: false
        autoUpdateSchemas: false
        allowNewBigQueryFields: false
        allowBigQueryRequiredFieldRelaxation: true
        schemaRetriever: com.wepay.kafka.connect.bigquery.retrieve.IdentitySchema
        bufferSize: 100
        maxWriteSize: 100
        tableWriteWait: 1000
        timestamp: UTC
        bigQueryPartitionDecorator: false
        project: wmt-gcp-promise-engine-prod
        defaultDataset: gtp_promise_engine_data
        keyfile: secret.ref://svc-promise-engine-creator-key.json
        bigQueryPartitionDecorator: false
```

Listing 6: Configuraciones del Connector

- **connector.class**: Se establece en `com.wepay.kafka.connect.bigquery.BigQuerySinkConnector`, lo que significa que se está usando el conector de BigQuery proporcionado por WePay.
- **max.batch.size**: Se establece en 100 para limitar el tamaño del lote de mensajes que se procesan en un solo request a BigQuery.

- **max.retries**: Se establece en 10 para limitar el número de intentos de reenvío en caso de fallo.
- **tasks.max**: Se establece en 5 para controlar el número máximo de tareas que se pueden ejecutar en paralelo para este conector.
- **topics**: Se establece en `cl-lastmile-promise-engine-slots-reserved` para indicar el nombre del topic de Kafka del que se obtendrán los datos.
- **autoCreateTables**, **autoUpdateSchemas**, **allowNewBigQueryFields**: Todos se establecen en `false` para evitar cambios automáticos en las tablas y esquemas de BigQuery.
- **allowBigQueryRequiredFieldRelaxation**: Se establece en `true` para permitir relajaciones en los campos requeridos de BigQuery.
- **schemaRetriever**: Se establece en `com.wepay.kafka.connect.bigquery.retrieve.IdentitySche` lo que significa que se está utilizando la clase IdentitySchemaRetriever proporcionada por WePay para la recuperación del esquema.
- **bufferSize** y **maxWriteSize**: Se establecen en 100 para limitar el número de registros que se pueden almacenar en el buffer y que se pueden escribir en BigQuery en un solo request, respectivamente.
- **tableWriteWait**: Se establece en 1000ms para controlar el tiempo de espera entre escrituras consecutivas a BigQuery.
- **timestamp**: Se establece en `UTC` para mantener una zona horaria coherente en los datos.
- **project** y **defaultDataset**: Se configuran para indicar el proyecto y el conjunto de datos predeterminado en BigQuery.
- **keyfile**: Se establece en `secret.ref://svc-promise-engine-creator-key.json` para especificar la ubicación de la clave privada usada para la autenticación en BigQuery.

Estas configuraciones permiten el funcionamiento adecuado del sistema en términos de eficiencia, seguridad y coherencia de datos, y están diseñadas para trabajar de la mano con las necesidades y limitaciones de las fuentes de datos y la infraestructura de BigQuery.

### 3.7.3. Configuración KITT de Kafka Connect

KITT es una herramienta desarrollada por Walmart diseñada para facilitar la transición de aplicaciones a un entorno de nube listo para producción dentro del ecosistema

WCNP, fundamentada en experiencias reales con Kubernetes. Además de brindar una implementación GitOps específica para Walmart, KITT está respaldado por herramientas como Concord, Looper y Artifactory, y se integra con recursos empresariales como GitHub, Slack, Microsoft Teams y otros. Esta plataforma no solo automatiza el despliegue de aplicaciones, sino que también incorpora funcionalidades como despliegues temporales, comunicaciones de pipeline y pruebas integradas, todo con una mínima configuración inicial.

En el archivo de configuración de KITT de Kafka Connect, se definen ciertas configuraciones globales en el ámbito de las métricas. La sección `remoteWriteSampleLimit` especifica un límite de 150 para las muestras de escritura remota, controlando así el número de muestras enviadas a un almacenamiento remoto en una sola transmisión.

La sección `whitelist` es particularmente importante, ya que lista las métricas de Prometheus que se desea exponer. Estas métricas brindan información valiosa sobre el rendimiento y la salud de Kafka Connect. Por ejemplo, las métricas que terminan en `avg` ofrecen una idea del rendimiento medio, mientras que aquellas que terminan en `total` proporcionan un conteo acumulado de ciertos eventos, como intentos de inicio, éxitos y fallos. Las métricas listadas abarcan desde la cantidad activa promedio de registros en las tareas de Kafka Connect hasta métricas relacionadas con el inicio de los conectores y tareas, como se especifica a continuación:

- `kafka_connect_sink_task_metrics_sink_record_active_count_avg`: Esta métrica proporciona el promedio de registros activos en las tareas del flujo de datos de Kafka Connect. Es útil para monitorear la carga de trabajo actual y cómo se distribuyen los registros entre diferentes tareas.
- `kafka_connect_sink_task_metrics_offset_commit_completion_total`: Representa el conteo acumulado de confirmaciones de offset que se han completado con éxito en las tareas del flujo. Un aumento súbito puede indicar un alto tráfico, mientras que un estancamiento puede señalar problemas en la confirmación de offsets.
- `kafka_connect_connect_worker_metrics_connector_startup_attempts_total`: Representa el número total de intentos para iniciar conectores en Kafka Connect. Si hay muchos intentos sin un correspondiente éxito, podría indicar problemas en la configuración del conector o en su integración.
- `kafka_connect_connect_worker_metrics_connector_startup_success_total`: Indica cuántas veces los conectores han sido iniciados con éxito. Idealmente, este número debería ser cercano al de intentos, lo que indicaría una tasa de éxito alta.
- `kafka_connect_connect_worker_metrics_connector_startup_failure_total`: Refleja el número total de fallos al intentar iniciar conectores. Un número elevado aquí es motivo de preocupación y requiere una investigación más detallada.

- `kafka_connect_connect_worker_metrics_task_startup_attempts_total`: Muestra el número total de intentos para iniciar tareas en Kafka Connect. Al igual que con los conectores, muchos intentos sin éxito pueden indicar problemas.
- `kafka_connect_connect_worker_metrics_task_startup_success_total`: Indica cuántas veces las tareas han sido iniciadas con éxito en Kafka Connect.
- `kafka_connect_connect_worker_metrics_task_startup_failure_total`: Representa el número total de fallos al intentar iniciar tareas. Al igual que con los conectores, un valor elevado en esta métrica es una señal de alarma.

Estas métricas son vitales para entender y monitorear la salud y el rendimiento de Kafka Connect. Una vigilancia regular y una respuesta rápida a las anomalías pueden garantizar un funcionamiento óptimo y prevenir posibles problemas.

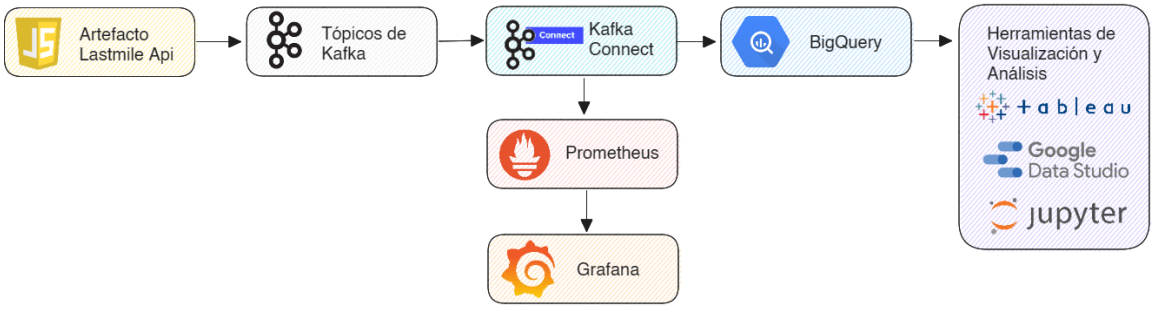



Figura 8: Arquitectura tomando en cuenta la pieza Kafka Connect.
Fuente: Elaboración Propia.

Al analizar detalladamente la componente de Kafka Connect, resulta esencial incorporar su monitoreo en la arquitectura final, tal como se muestra en la Figura 8.

### 3.7.4. Implementación de un Tablero de Monitoreo Basado en Métricas Prometheus


Figura 9: Dashboard de Monitoreo Kafka Connect.
Fuente: Elaboración Propia.

A fin de optimizar el monitoreo de este componente, se desarrolló un dashboard en Grafana (ver figura 9), capitalizando las métricas recolectadas y expuestas por Prometheus. Esta combinación permite una visualización efectiva y en tiempo real del rendimiento de Kafka Connect.

## 3.8. Análisis detallado de la visualización del tablero de monitoreo

### 3.8.1. Implementación y Utilidad de las Visualizaciones de Flujo de Mensajes

Una característica principal del tablero de monitoreo es su capacidad para visualizar el flujo de mensajes a través de Kafka y su envío posterior a BigQuery para cada tópico al que está conectado el conector. Esta visibilidad es crítica para entender el funcionamiento del sistema y garantizar su correcta funcionalidad.

Para representar de manera efectiva estos flujos de datos, se recurrió a la utilización de dos tipos de visualizaciones de Grafana.

En primer lugar, se encuentra un *Pie Chart* o gráfico circular. Este gráfico proporciona una visión resumida del total de mensajes que se procesan en cada tópico, permitiendo a

simple vista una comparación entre los diferentes tópicos y una comprensión intuitiva de la proporción del volumen total de mensajes que cada uno de ellos representa.

Adicionalmente, se incorporó un gráfico de *Time-Series* o series de tiempo. Este tipo de visualización permite observar en detalle el flujo de mensajes a lo largo del tiempo, aportando un contexto temporal a las métricas y ofreciendo una percepción más completa y granular del rendimiento y la salud de cada tópico. De esta forma, se facilita la identificación de patrones y tendencias, así como la detección temprana de posibles anomalías o problemas.

Estas visualizaciones, al combinar una visión resumida y detallada del flujo de mensajes, ayudan a garantizar la salud y la eficiencia de los tópicos. La igualdad en el flujo de consumo y de envío de mensajes para cada tópico es un indicador de salud y estabilidad del sistema, y estas visualizaciones facilitan la monitorización y verificación de este aspecto crítico. Con ello, se asegura un monitoreo efectivo y la capacidad de reaccionar rápidamente ante cualquier desequilibrio o anomalía.


Figura 10: Panel ”Sink Records Metrics”.
Fuente: Elaboración Propia.

### 3.8.2. Implementación de Métricas Singlestat para Control de Conectores y Tareas

En una sección complementaria del tablero de monitoreo, se implementaron métricas Singlestat para obtener un resumen cuantitativo de los procesos de inicialización y fallos que se produzcan en cada conector y tarea de Kafka Connect. Este tipo de visualización proporciona una visión instantánea del estado y comportamiento de estos elementos críticos en el sistema.

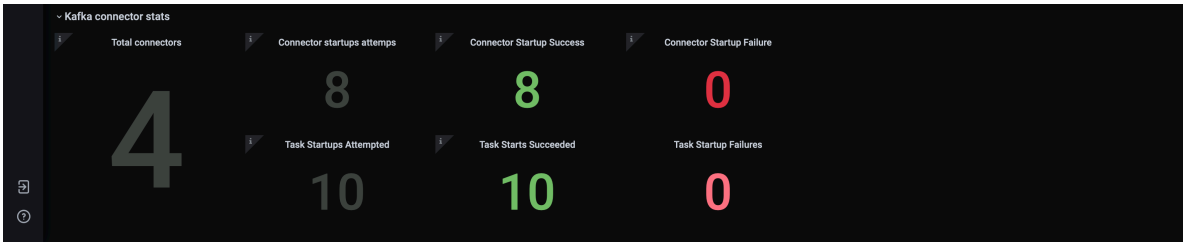



Figura 11: Panel Kafka Connector Stats.
Fuente: Elaboración Propia.

El uso de Singlestat es particularmente útil para monitorear la salud y el estado operativo de cada conector y tarea de Kafka Connect. La visualización proporciona un resumen en tiempo real del rendimiento de estos componentes, permitiendo identificar rápidamente cualquier fallo o anomalía que pueda ocurrir durante los despliegues del artefacto o en su funcionamiento posterior y así tener un entendimiento instantáneo y claro de su estado operativo.

## 3.9. Visualización y análisis de datos

### 3.9.1. Tableau: Diseño y Funcionalidades del Dashboard Geográfico

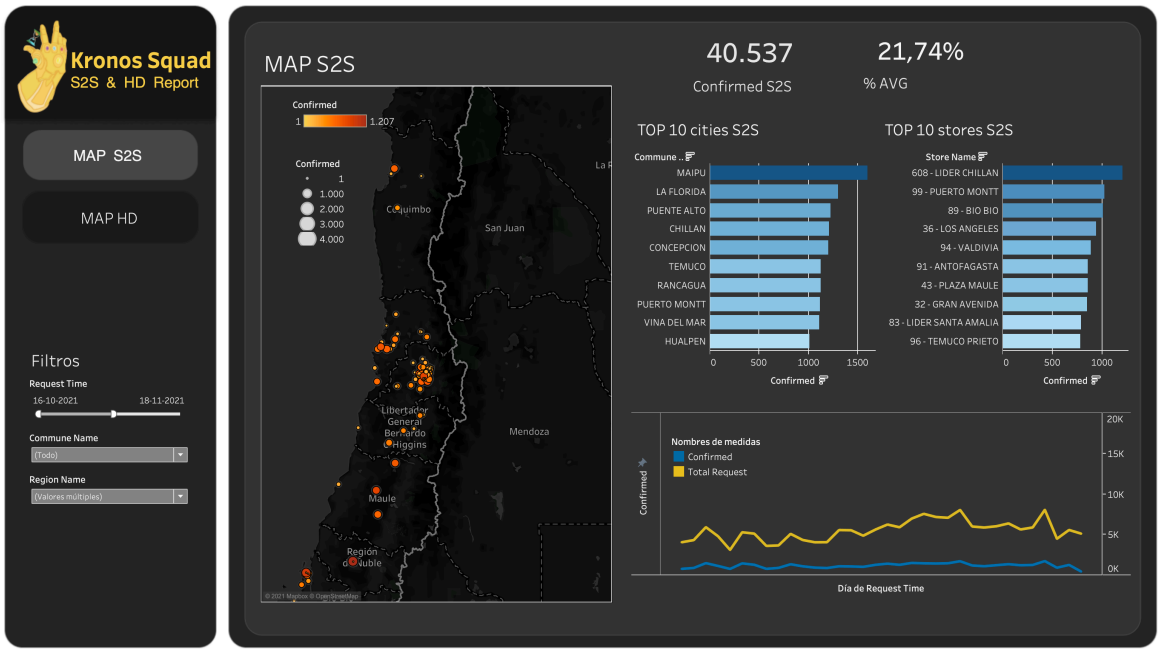



Figura 12: Dashboard Geográfico Retiro en Tienda.
Fuente: Elaboración Propia.

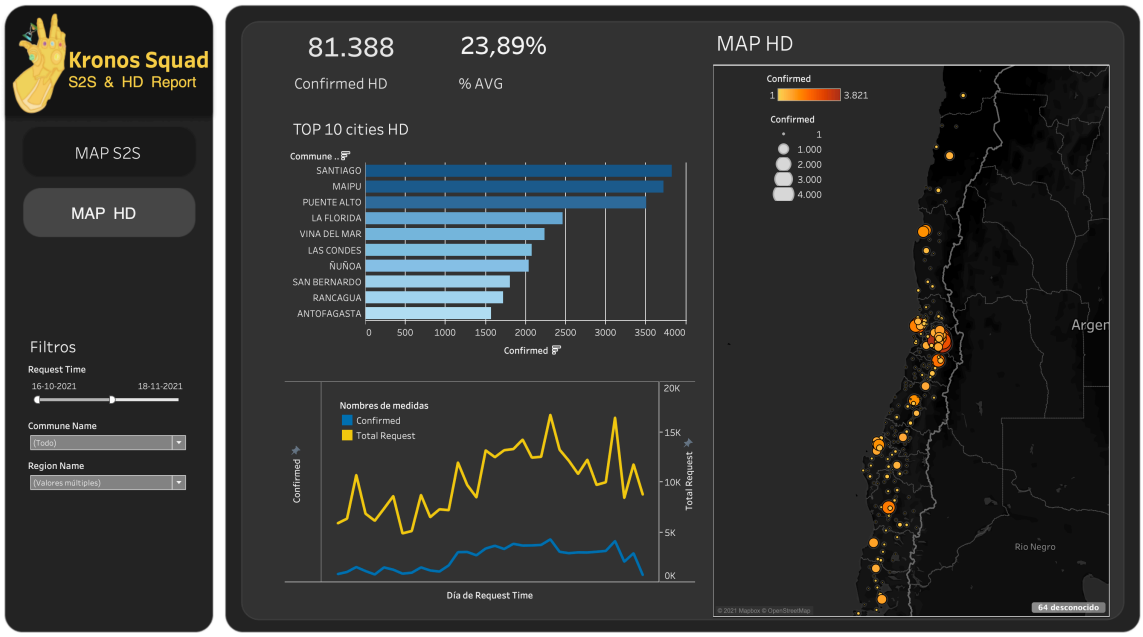



Figura 13: Dashboard Geográfico Home Delivery
Fuente: Elaboración Propia.

En el núcleo de la estrategia analítica se encuentra el panel de control geográfico desarrollado en Tableau, que combina una representación detallada con insights prácticos sobre las operaciones y el rendimiento de ventas. Este panel interactivo es instrumental para evaluar el desempeño de las entregas, especialmente las modalidades ”Ship to Store”(S2S) y ”Home Delivery”.

El diseño del panel se bifurca estratégicamente en dos vistas. La primera está orientada a ofrecer un análisis exhaustivo del servicio de entrega S2S a nivel nacional (Ver figura 12). En contraste, la segunda se enfoca en detallar el desempeño del servicio Home Delivery (ver figura 13).

Un componente esencial en este panel es el mapa de densidad. No solo representa gráficamente las ventas en diferentes localidades, sino que también destaca las ciudades con el mayor volumen de confirmaciones, un indicador directo de las ventas efectuadas. Esta herramienta visual es particularmente útil para identificar zonas de alta demanda, facilitando la optimización de las estrategias logísticas y de marketing.

El análisis temporal incorporado, visualizado mediante un gráfico de series de tiempo, ofrece una panorámica de la evolución de solicitudes de ventanas y las respectivas confirmaciones. Esta vista dinámica es invaluable para detectar tendencias y patrones emergentes en la actividad comercial.

En la sección superior del panel, se ofrece un resumen cuantitativo, mostrando el número

total de pedidos confirmados y su relación porcentual con las solicitudes de ofertas generadas. Este dato proporciona una evaluación instantánea de la eficiencia de las operaciones de entrega.

Finalmente, y para reforzar la perspectiva estratégica, en la vista S2S se han resaltado las 10 ciudades con el mayor volumen de confirmaciones, así como las 10 tiendas líderes en retiros de pedidos. Estos insights son cruciales al planificar y evaluar la eficiencia de los puntos de servicio.

### 3.9.2. Tableau: Análisis y Características del Dashboard de Resumen del Estado de las Ventanas SRS

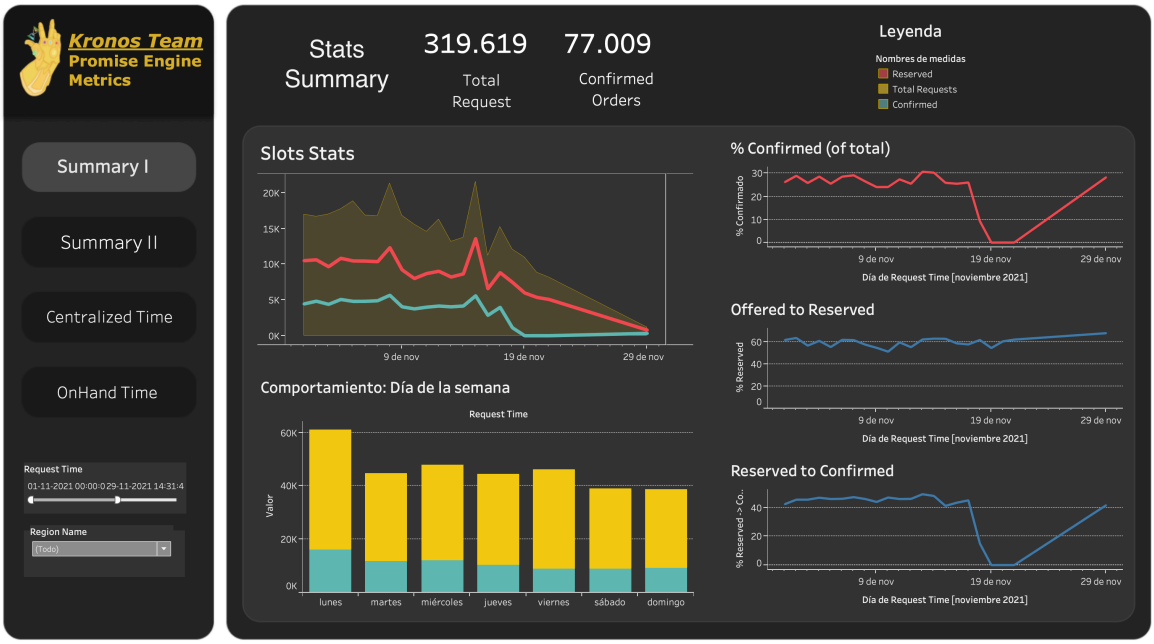



Figura 14: Dashboard de resumen I.
Fuente: Elaboración Propia.

El Dashboard de Resumen del Estado de las Ventanas SRS fue diseñado para proporcionar un panorama integral sobre la situación actual de las ventanas de despacho en la operación. Este instrumento permite analizar de manera efectiva y precisa las interacciones de los clientes con las ventanas de despacho.

El primer panel del dashboard (ver figura 14) cuenta con tres secciones claves, la primera es **Slots Stats**, donde se presenta una serie de tiempo con una representación gráfica de distintas métricas. El sombreado amarillo representa el total de solicitudes de ventanas, mientras que las líneas roja y celeste denotan las reservas realizadas y la cantidad de

confirmaciones respectivamente. Este gráfico permite seguir de cerca la progresión y la evolución de la demanda, las reservas y las confirmaciones a lo largo del tiempo.

En el siguiente panel, se muestra un gráfico de barras que ilustra el comportamiento de las solicitudes de ventanas durante los días de la semana. Este gráfico brinda una comprensión clara de cuándo los clientes suelen consultar más las ventanas de despacho.

Finalmente, el último panel se centra en los indicadores de conversión de ventas. Se muestra el porcentaje confirmado del total, que representa cuántas ventanas fueron confirmadas con respecto a las ofertas realizadas. Además, se exhibe el porcentaje de conversión de la oferta de ventana a reserva y el porcentaje de conversión de reserva a confirmación. Estos indicadores son de gran importancia para evaluar la efectividad de las operaciones y la satisfacción del cliente con las ventanas de despacho ofertadas.

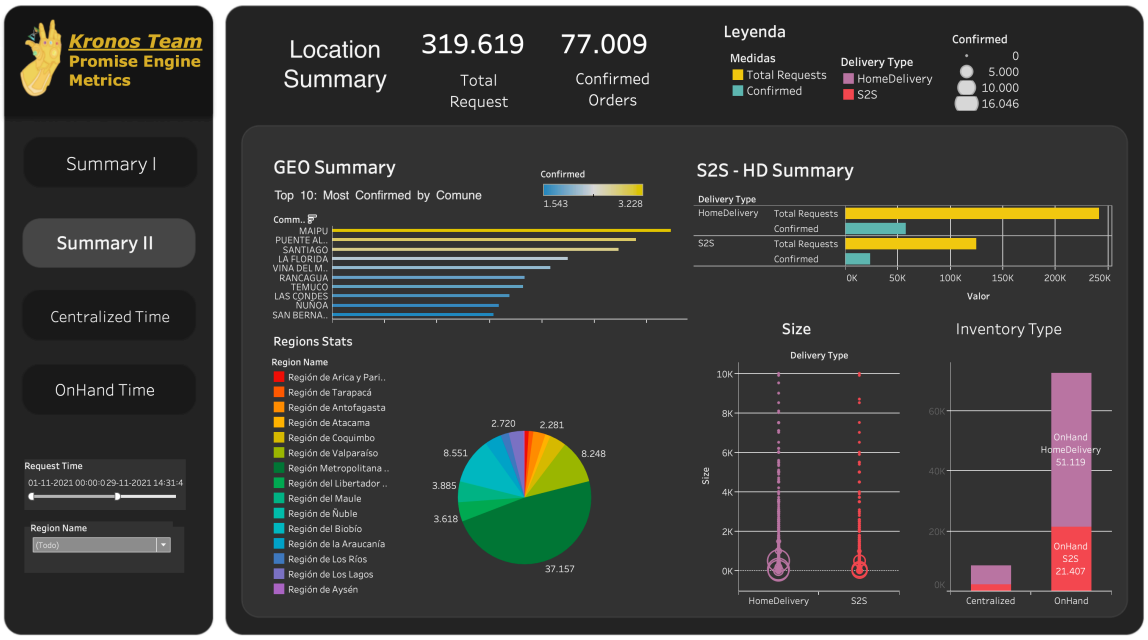



Figura 15: Dashboard de resumen II
Fuente: Elaboración Propia.

La segunda vista del Dashboard de Resumen del Estado de las Ventanas SRS proporciona una imagen detallada y segmentada de las operaciones en función de la ubicación geográfica y del tipo de entrega (ver figura 15).

La sección del resumen geográfico destaca las diez ciudades donde se registran el mayor número de confirmaciones de ventanas. Este análisis permite entender las regiones de mayor demanda. Acompañando este dato, se presenta un gráfico circular (*Pie Chart*) que desglosa el porcentaje de confirmaciones correspondientes a cada una de estas regiones. Esto ofrece una imagen clara de la distribución de las operaciones a nivel geográfico.

Paralelamente, la segunda sección de esta vista se centra en el análisis del comportamiento de las ventanas de tipo *Home Delivery* y *S2S*. Primero, se expone un gráfico de barras (*Bar Plot*) que compara la cantidad de ofertas de ventanas presentadas a los clientes y la cantidad de confirmaciones obtenidas. Esta comparación permite evaluar la aceptación y eficacia de las ofertas de ventanas.

Posteriormente, para examinar el comportamiento según el tamaño de la orden, se utiliza una vista circular (*Circle View*). En este gráfico, el tamaño del círculo representa la cantidad de confirmaciones, lo que permite una rápida identificación visual de las órdenes más frecuentes y su volumen de confirmaciones.

Finalmente, se detalla la cantidad de confirmaciones obtenidas en cada tipo de envío, clasificadas según el tipo de inventario. Este dato es crucial para entender las preferencias de los clientes y el rendimiento de las diferentes opciones de entrega.

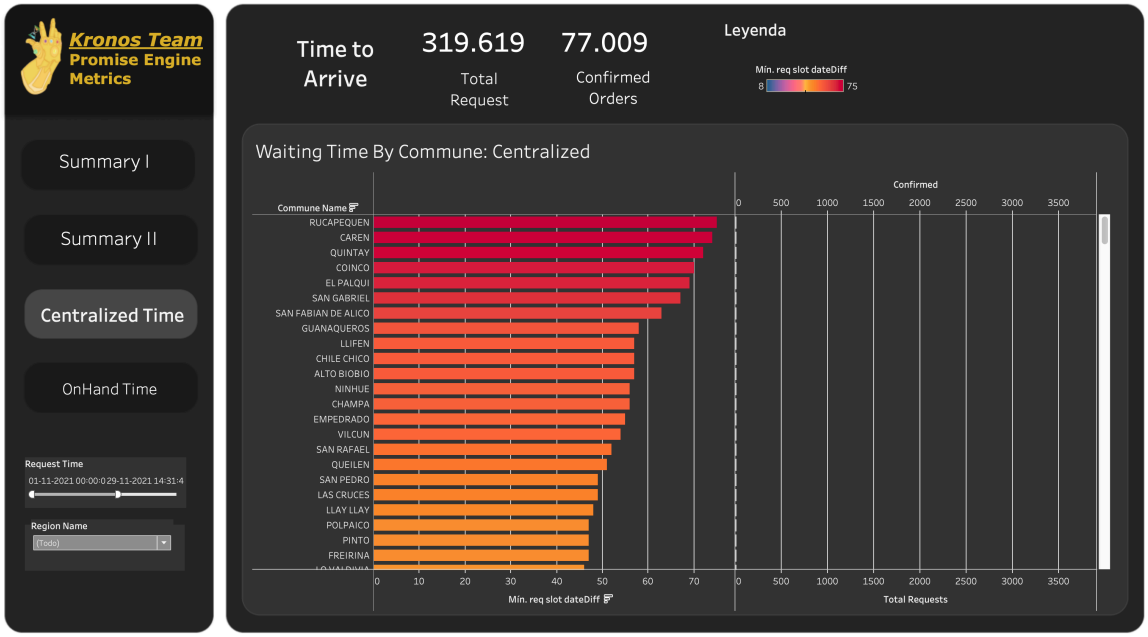



Figura 16: Dashboard de resumen III
Fuente: Elaboración Propia.

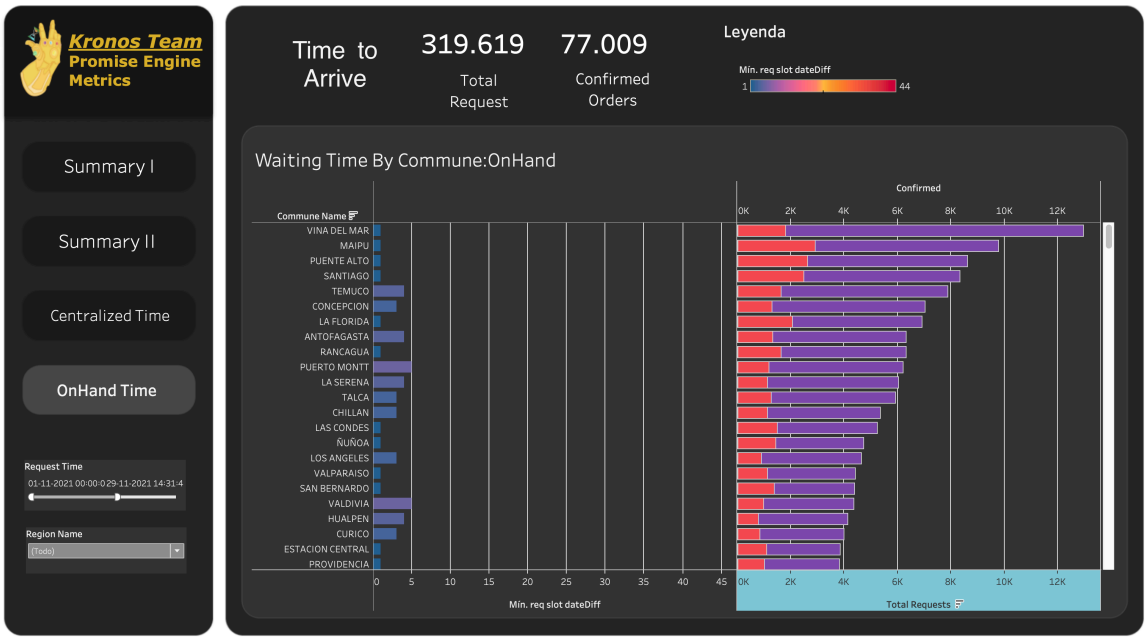



Figura 17: Dashboard de resumen IV
Fuente: Elaboración Propia.

La siguiente vista del Dashboard de Resumen del Estado de las Ventanas SRS proporciona una perspectiva detallada de la oferta de ventanas más próxima disponible en cada ciudad, así como de la cantidad de ofertas y confirmaciones registradas (ver figuras 16 y 17).

El panel se divide en dos secciones principales para facilitar la interpretación de la información.

En la sección izquierda, se ofrece un análisis temporal que relaciona el día en que se realiza la solicitud de oferta de ventanas y la fecha de despacho más próxima disponible. Este dato se traduce en la cantidad de días que un cliente tendría que esperar para recibir su paquete una vez realizada la solicitud. Este indicador es crucial para evaluar la eficacia y rapidez del servicio de entrega, así ayuda a identificar posibles áreas de mejora en la gestión de las ventanas de despacho.

En contraste, la sección derecha del panel se centra en la cantidad total de ofertas de ventanas presentadas y confirmaciones obtenidas en cada ciudad. Esta información es esencial para comprender el rendimiento de las operaciones en cada región, identificar patrones y tendencias en la demanda.

En conjunto, esta vista del dashboard permite tener un panorama claro del desempeño de las ofertas de ventanas con relación a la proximidad de entrega y su aceptación por parte de los clientes en diferentes regiones.

### 3.9.3. Tableau: Análisis del Costo de las Ventanas de Despacho

El siguiente dashboard se centra en proporcionar una visión detallada y desglosada del costo asociado con las ventanas de despacho.

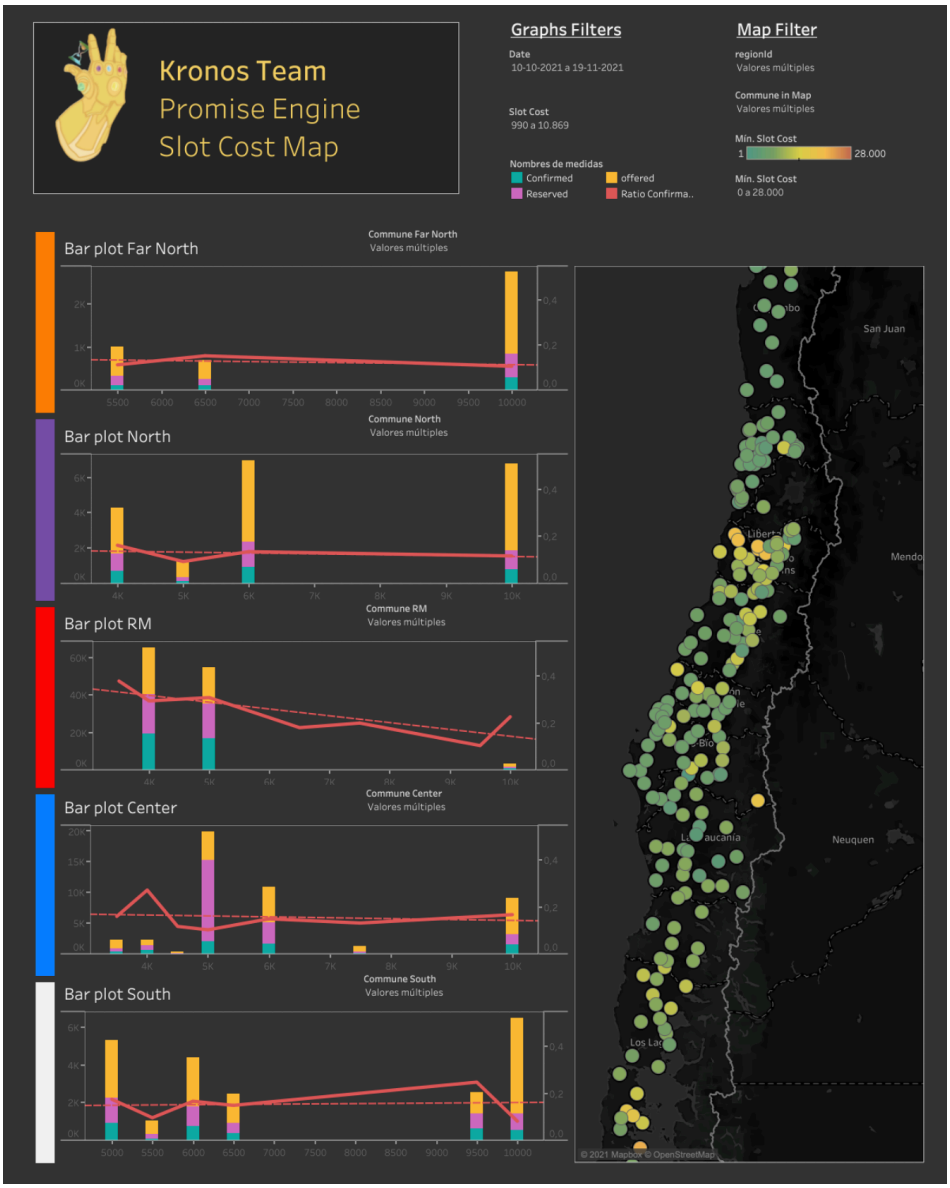



Figura 18: Dashboard Geográfico Costo de ventanas.
Fuente: Elaboración Propia.

El dashboard se divide en dos secciones principales para ofrecer una vista más comprensiva y precisa de los costos.

En la sección izquierda del panel, se dispone de gráficos de barras que representan

la cantidad de ofertas, reservas y confirmaciones en función del costo del envío. Esta información se desglosa para cada zona del país, permitiendo una comparativa directa y un análisis detallado de las variaciones de costos y su impacto en las reservas y confirmaciones en las diferentes zonas.

Por otro lado, la sección derecha del panel presenta un mapa interactivo. Cada comuna y localidad del país en donde se ofrece envío se representa con un círculo, cuyo color varía en función del costo medio de envío en esa zona. Esta representación visual facilita la identificación de las áreas con mayor y menor costo, proporcionando una rápida comprensión de la distribución geográfica de los costos de envío.

En conjunto, este dashboard de costo de ventanas permite una comprensión más profunda y completa de los costos de las operaciones de entrega. A través de la combinación de gráficos de barras y visualizaciones geográficas, se pueden identificar patrones, tendencias y áreas de mejora en la gestión del costo de las ventanas de despacho.

### 3.9.4. Tableau: Dashboard detalle según tipo de envío en regiones


Figura 19: Dashboard detalle tipo de envío.
Fuente: Elaboración Propia.

Este dashboard proporciona un análisis más granular y específico de las diferentes zonas del país, extendiendo y profundizando la información presentada en la figura 12 que pertenece

segunda vista del dashboard de resumen.

Aquí, se ha replicado la visualización de tipo *Circle View* y el detalle según el tipo de envío, manteniendo las mismas métricas y criterios de visualización para una interpretación consistente y coherente de los datos a lo largo de los diferentes paneles. Sin embargo, a diferencia del panel de resumen, donde se proporciona una vista general para todo el país, en este caso se ofrece un desglose detallado por cada zona individualmente.

De esta manera, se facilita la identificación y análisis de las particularidades y tendencias específicas de cada región.

### 3.9.5. Jupyter Notebook: Análisis de Preferencias de Ventana

En el contexto de la validación de las hipótesis y en alineación con el objetivo HDU-06 (véase Tabla 9), se implementó un análisis detallado mediante Jupyter Notebook, utilizando Python como lenguaje principal.

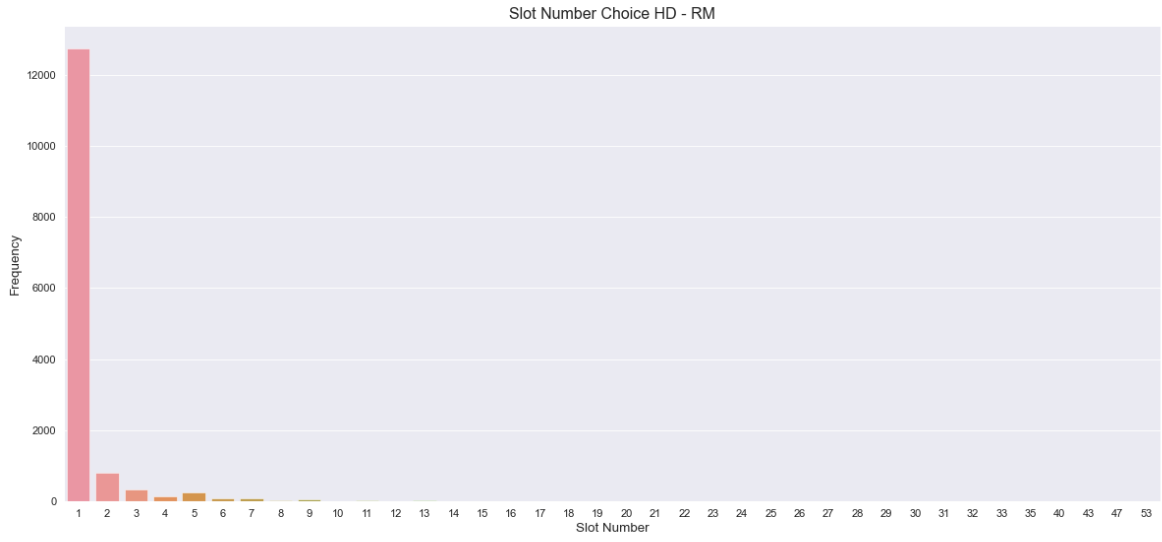



Figura 20: Gráfico reporte Elección de Ventana para Home Delivery en la Región Metropolitana.

Fuente: Elaboración Propia.

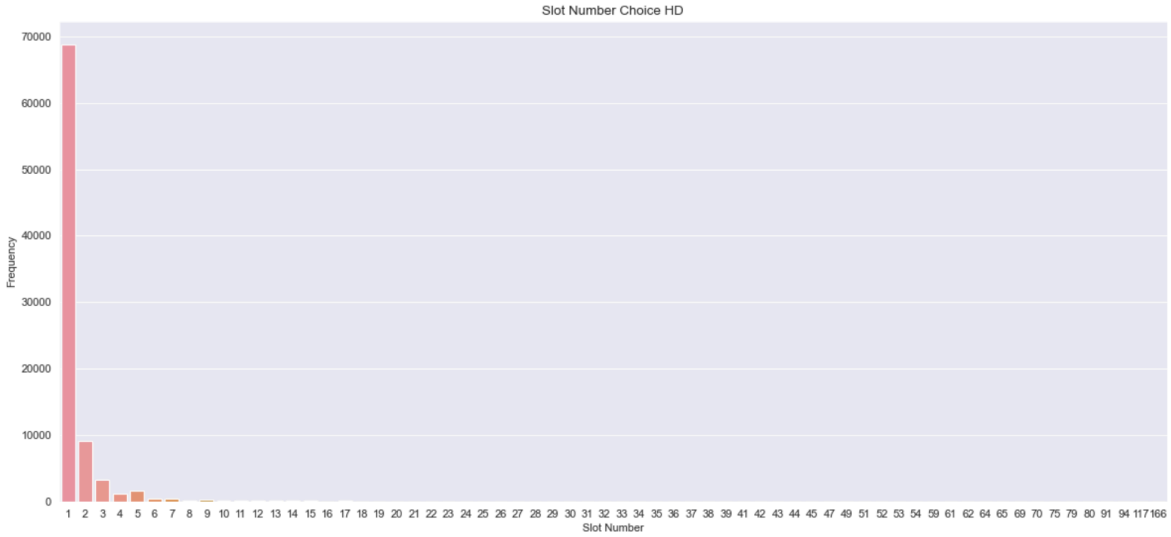



Figura 21: Gráfico reporte Elección de Ventana para Home Delivery.
Fuente: Elaboración Propia.

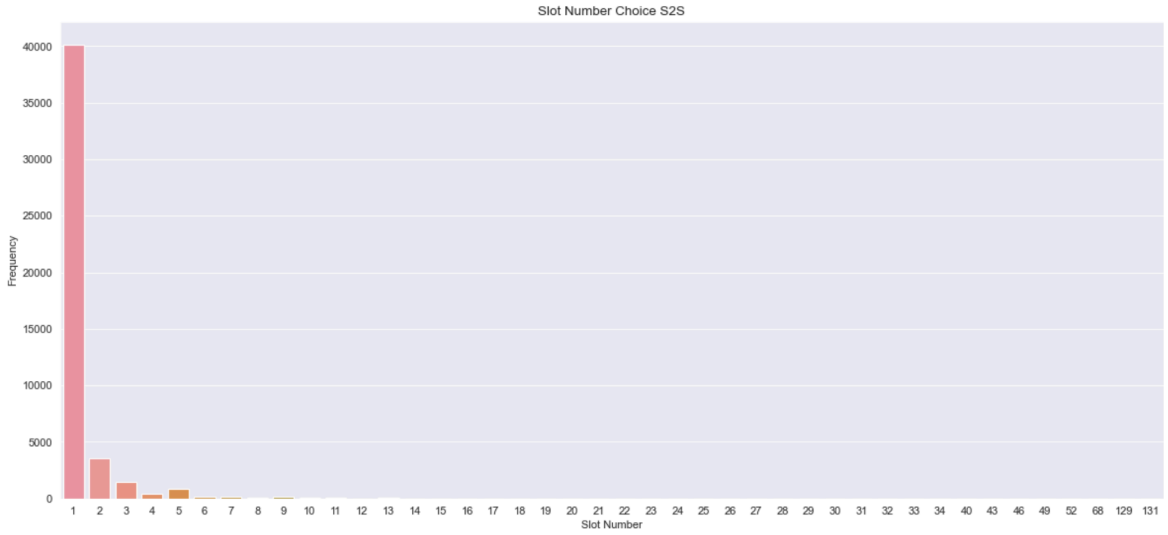



Figura 22: Gráfico reporte Elección de Ventana para S2S.
Fuente: Elaboración Propia.

Este reporte se benefició de una vista previamente estructurada en BigQuery, en la cual se añadió un atributo que identifica la ventana seleccionada de entre todas las opciones ofrecidas al cliente durante la fase de confirmación. Esta adición provee una visión más clara sobre las inclinaciones y decisiones de los clientes en cuanto a la elección de ventanas. Además, al presentar esta información en orden cronológico, es factible determinar si el cliente tendió a seleccionar la ventana más inmediata, interpretado como la primera opción disponible.

El análisis visual se despliega en las figuras 18, 19 y 20. En ellas, se muestra un *Bar Plot* que denota la frecuencia de elección de ventanas basada en su orden de proximidad. Para claridad, en esta representación, el número 1 simboliza la ventana más inmediata o primera opción dada, el número 2 representa la siguiente ventana en términos de cercanía, y así consecutivamente. Este enfoque gráfico brinda una perspectiva intuitiva sobre las tendencias y patrones en las decisiones de los clientes al seleccionar ventanas propuestas.

## 3.10. Adaptación a DataStudio


Figura 23: Dashboards análogos a los de Tableau
Fuente: Elaboración Propia.


Figura 24: Dashboard
Fuente: Elaboración Propia.

Dentro de Walmart, no todos los miembros del equipo tienen acceso a las herramientas de Tableau. Por ello, se buscó una alternativa que permitiera compartir ciertos informes con una audiencia más amplia. Se optó por Datastudio para esta tarea. Aunque se intentó que los dashboards en Datastudio fueran lo más similares posible a los creados en Tableau, las limitaciones inherentes de Datastudio impidieron replicarlos en su totalidad. A pesar de ello, se lograron recrear principalmente los dashboards resumidos, se basó en los resúmenes 1 y 2 de Tableau (ver figuras 14 y 15). Además, se diseñó una vista específica para cada región del país, mostrando un resumen de métricas como tipo de envío, tamaño de orden y tipo de inventario.

En una adición relevante, la última vista de Datastudio incorporó los KPIs de Same Day Picking, Next Day Picking y Next Day Delivery. Estos indicadores son cruciales para las operaciones, ya que el equipo debe implementar diversas configuraciones y adaptaciones logísticas para optimizar los tiempos de entrega.

# CAPÍTULO 4
# VALIDACIÓN DE LA SOLUCIÓN

El propósito inicial del proyecto consistía en completar todas las historias de usuario para finalizar con la entrega de un sistema integral de análisis de datos. Esta validación se planteó para asegurar la efectividad de la solución propuesta.

## 4.1. Validación de Hipótesis

Al inicio del proyecto, se plantearon varias hipótesis para validación. Aunque los tableros de control (dashboards) creados ofrecieron cierto nivel de verificación a estas hipótesis, es importante destacar que la confirmación y la precisión de las mismas pueden estar sujetas a cambios con la incorporación de datos adicionales. Este cambio puede deberse a factores tales como campañas comerciales en curso o incidentes en la plataforma online, entre otros. Aunque las hipótesis se basan principalmente en la condición de las ventanas de despacho, no siempre se podrá tener una visión completa del panorama como lo sería, teniendo disponible otros datos como el costo del producto, los datos de marketing, los datos del cliente, entre otros.

### 4.1.1. Hipótesis de Costo de Slots

**El costo de entrega influye en la decisión de compra del cliente.**
De acuerdo con el tablero de control de Costos (ver figura 18), la tendencia general indica una pendiente negativa. Esta observación sugiere que el costo de entrega puede ser un factor determinante en la decisión final de compra del cliente. Cabe destacar que este mismo análisis fue compartido con el equipo de datos de Walmart, llegando a la misma conclusión, pero con más detalle al ellos tener el stack de datos completo de Walmart, posterior a esto se implementó una iniciativa que ofrece envío gratuito en compras que superen los $15.000.

**El costo de entrega afecta la elección del tipo de entrega seleccionada por el cliente (s2s).**
Se observa que las áreas con mayores ventas de home delivery también registran una gran cantidad de ventas S2S. Al alejarse de la zona centro del país, generalmente el despacho es más costoso, con esto en mente se puede encontrar ejemplos donde las compras S2S de algunas ciudades superan a las de las ciudades con mayor densidad de población. Un caso ilustrativo de esta situación podría ser Chillán. A pesar de contar con aproximadamente 160.000 habitantes, presenta más ventas bajo la modalidad de retiro en tienda que una ciudad como Viña del Mar, la cual tiene casi el doble de población. Este comportamiento es coherente con el hecho de que, a medida que se aumenta la distancia desde los centros de distribución, el costo del slot de entrega tiende a ser mayor, por lo tanto, algunos prefieren

retirar en tienda de manera gratuita. Aun así, no se puede asegurar completamente que este sea el caso, también puede ser que dado que es una región con más zonas rurales, no se ofrece despacho a domicilio para todas las direcciones.

Teniendo en cuenta la información presentada, no fue posible validar plenamente la hipótesis basándose únicamente en los datos de las ventanas de despacho; por ende, el resultado es no concluyente.

### 4.1.2. Elección de Slot

**Los clientes tienden a seleccionar la primera franja de entrega disponible (HD).**
Según los datos recopilados y visualizados en el reporte generado en Python (ver figuras 20, 21 y 22), se evidencia una tendencia de los clientes a seleccionar la fecha de entrega más próxima. A pesar de ofrecer un rango de fechas amplio para dar flexibilidad al cliente, parece ser que la opción más atractiva para ellos es recibir su compra lo antes posible, sin tomar en cuenta su disponibilidad personal.

### 4.1.3. Tamaño del producto

**Los clientes tienden a preferir HD cuando el tamaño del producto es grande.**
Para verificar esta hipótesis, se empleó el tablero de resumen (ver figura 16) y el tablero de detalle (ver figura 19). Se puede apreciar que el número de confirmaciones es similar en ambas modalidades de entrega, sin embargo, se nota que para la modalidad S2S, a medida que el tamaño del producto aumenta, el número de confirmaciones disminuye. Al contrario, en las zonas alejadas del centro, se observa una mayor cantidad de confirmaciones para productos grandes en la modalidad de Home Delivery.

Aunque la información mostrada tiende a respaldar la hipótesis, sería beneficioso acceder a los datos de los productos adquiridos por orden. Esto permitiría confirmar si efectivamente son artículos de gran tamaño, y no simplemente múltiples productos pequeños que acumulan una orden grande.

### 4.1.4. Indicadores de rendimiento deseables

La representación de la conversión en el flujo de venta se muestra en las figuras 14 y 23. Esta visualización permite evaluar el estado de las ventanas de despacho, facilitando el monitoreo diario de su comportamiento. Con ello, se pueden apreciar los impactos de las campañas en el flujo de venta, así como de incidentes relacionados con las ventanas.

En lo que concierne a la segmentación por Tipo de Inventario, se dispone de dashboards

específicos que permiten analizar el comportamiento según el tipo de inventario (ver figuras 16, 17 y 19). Es especialmente relevante para inventarios centralizados observar el tiempo de espera que un cliente podría enfrentar. Los esfuerzos que se realizan para minimizar estos tiempos de espera no solo dependen de la empresa, sino también del vendedor, quien tiene la responsabilidad de entregar su producto a Lider.

En lo que se refiere a esto, la figura 24 muestra la implementación de los KPIs para 'same day delivery', 'next day picking' y 'same day picking'. La capacidad de filtrar por comuna es de gran utilidad para evaluar la efectividad de las configuraciones operativas. El objetivo es incrementar la cantidad de ventanas que ofrecen esta modalidad, en especial fuera de la región metropolitana, donde es esencial competir con los tiempos de entrega de otras compañías.

## 4.2. Entrevistas

Con el propósito de evaluar el impacto y valor de la solución propuesta, se consideró esencial recopilar las opiniones de aquellos trabajadores directamente afectados por ella. Las entrevistas se diseñaron como una herramienta para validar la pertinencia y efectividad de la solución en el contexto real de trabajo.

### 4.2.1. Perspectiva del negocio y operaciones

Para obtener una visión más completa, se solicitó la opinión del dueño del producto. Destaca el valor que los datos brindan al negocio, facilitando discusiones e iniciativas basadas en datos. A pesar de que estos datos deben ser corroborados por el equipo de análisis, la existencia de un tablero propio agiliza la conversación inicial y abre la puerta para convertir propuestas en realidad.

Además, la disponibilidad de datos en un datamart permite al equipo de datos tener una imagen más completa del ecosistema de Walmart CL. Si necesitan más informes, tienen a su disposición los tres eventos del producto Promise Engine.

Al dialogar con el jefe de operaciones del centro de distribución, este afirmó que el mayor valor de la solución es el almacenamiento de datos. Los tableros permiten validar teorías y justificar acciones. Sin embargo, en un mundo de constantes cambios como el ecommerce y las multinacionales, tener acceso a información en tiempo real es lo que más beneficia.

Se puede concluir que, aunque los tableros son útiles para casos específicos, la disponibilidad de datos permite trabajar con ellos, incluso si un tablero deja de ser útil debido a cambios en el negocio.

### 4.2.2. Perspectiva de los desarrolladores

Desde una visión más técnica y dialogando con los desarrolladores del producto, valoran tener un sistema que permite un seguimiento continuo. Antes de implementar este sistema, resultaba difícil dar soporte a algo que no proporcionaba métricas de salud, ni ofrecía visibilidad sobre su estado. Con la nueva solución, pueden ver en tiempo real el consumo y el envío de datos a BigQuery, así como el estado de los conectores y la posibilidad de realizar un diagnóstico en caso de fallos. Además, la solución es compatible con el ecosistema GTP, lo que permite una migración sin el temor de perder datos al incorporarse a la nueva plataforma.

# CAPÍTULO 5
# CONCLUSIONES

## 5.1. Desarrollo y Reflexión sobre la Implementación de Business Intelligence en Walmart CL

El propósito primordial de este proyecto radicaba en instaurar un sistema para el análisis y monitoreo detallado de los datos vinculados a las ventanas de despacho, apoyándonse en una metodología ágil. Gracias a un minucioso levantamiento de requerimientos desde el inicio, y al diseño de spikes e historias de usuario, se pudo integrar al flujo de trabajo de Walmart de manera orgánica y fluida, logrando resultados en poco tiempo.

Dentro de los objetivos específicos, se destacaba la necesidad de forjar una solución de Business Intelligence específicamente para las ventanas de despacho. El diseño de la arquitectura, la creación del pipeline y el sistema de almacenamiento de datos se ejecutaron con notable éxito. Se consiguió establecer un flujo de datos integral desde la pieza de lastmile hasta las tablas en BigQuery, garantizando, además, observabilidad que fortalece el soporte por parte del equipo de desarrollo.

Los dashboards diseñados demostraron ser instrumentos vitales para la validación de hipótesis. No obstante, una vez testadas, se identificó que el ámbito de Business Intelligence es dinámico, y los dashboards deben adaptarse según las cambiantes necesidades organizacionales. Un claro ejemplo de esta adaptabilidad surgió cuando, tras la validación, se lanzó una campaña de envío gratuito al superar un monto específico de compra. Una visión más integral del proceso, como contar con la data de checkout para determinar el monto total, es esencial, enfocarse únicamente en las ventanas podría conducir a una interpretación sesgada. Es importante subrayar que no todas las hipótesis pueden validarse con la información disponible; aunque ciertos datos puedan respaldar una hipótesis, es fundamental contar con una perspectiva más amplia y detallada para confirmar su validez genuina.

Desde la perspectiva de ingeniería de datos, la solución diseñada mostró ser adaptable tanto en CL como en GTP, representando un valor añadido para el equipo. La eficiencia del pipeline y la facilidad para mantener herramientas como Kafka Connect y BigQuery, especialmente en la gestión y expansión de schemas, resaltan su robustez.

Este proyecto marcó un hito en Walmart CL, al ser el primero en adoptar tecnologías de GTP para una solución de datos integral. Este logro sentó precedentes para futuras migraciones. Si bien actualmente existe un equipo de Gobernanza de Datos encargado de esta área, este trabajo sirvió como un bloque fundamental previo a su transición hacia GTP, facilitando su tarea.

Sin embargo, no todo fue sencillo. Se desconocía ciertas particularidades burocráticas de las multinacionales, lo que a veces ralentizó procesos esperados. Estas trabas afectaron la agilidad, especialmente al no anticipar todas las dependencias externas de cada Historia de Usuario. Este aprendizaje hizo reconsiderar la estructura y atomización de las historias de usuario para otra oportunidad.

A nivel de visualización, la elección de herramientas fue esencial. Aunque inicialmente se trabajó con Tableau por su potencia analítica, se tuvo restricciones de acceso para algunos desarrolladores. Finalmente, se optó por migrar ciertos dashboards a DataStudio, una herramienta más accesible pero menos potente.

Hubiera sido enriquecedor contrastar Kafka Connect con opciones serverless como Dataflow. Sin embargo, dadas las tecnologías preexistentes y consideraciones de costos, Kafka Connect se erigió como la mejor opción, sin embargo, hubiese sido interesante poder realizar una prueba comparativa entre ambas.

En lo personal, esta experiencia resalta la sinergia crucial entre desarrolladores de software y analistas de datos. Ambos roles, al colaborar estrechamente, optimizan la implementación de soluciones. Conocer en profundidad las fuentes de datos y la analítica permite enfrentar desafíos, brindando soporte efectivo y reaccionando con agilidad ante incidentes al tener una mirada completa.

En conclusión, este proyecto no solo refuerza la interconexión entre desarrollo técnico y análisis de datos, sino que también pone de manifiesto la adaptabilidad y preparación necesaria para enfrentar las dinámicas cambiantes del comercio electrónico en un gigante como Walmart. Esta labor demuestra cómo, con las herramientas y enfoques adecuados, se pueden generar soluciones robustas y resilientes que beneficien a toda la organización.

## 5.2. Trabajos Futuros

Desde la perspectiva del desarrollo, una evolución propuesta para el Promise Engine involucra la incorporación de una o más componentes que se vinculen directamente con Kafka Connect, operando de manera autónoma a la API de las ventanas de despacho lastmile. Actualmente, el sistema opera como un monolito, lo que podría representar desafíos al configurar o escalar. A medida que lastmile crece, tanto los consumidores como los productores experimentan un incremento, lo que desencadena rebalanceos que pueden comprometer el rendimiento del artefacto. Específicamente, la gestión simultánea de la mensajería podría debilitar la eficacia del servicio durante eventos de alta demanda, como las campañas comerciales del tipo cyber.

En lo que respecta a la infraestructura de GTP, se observó que BigQuery superó en velocidad a Dremio. Esta constatación emergió luego de que el equipo de gobernanza de datos, al asumir los datos del Promise Engine, optase inicialmente por migrar todo

a Dremio. No obstante, al realizar una prueba comparativa de rendimiento entre ambas soluciones, BigQuery demostró ser más eficiente. Esta diferencia de performance llevó a reconsiderar la elección inicial y optar por BigQuery como solución de almacenamiento principal, reafirmando su relevancia y eficacia en la gestión de datos a gran escala, pero provocando que se deba realizar a futuro un nuevo movimiento de datos, esta vez de Dremio a BigQuery.

# REFERENCIAS BIBLIOGRÁFICAS

[Anderson, 2010] Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.

[Authors, 2021] Authors, P. (2021). Documentation.

[Banguero y Amaya, 2018] Banguero, E. y Amaya, J. (2018). Agile methodologies: Scrum vs. kanban. *Scrum y Kanban*, 1(1):1–5.

[Chiara Brocchi y Neiman, 2016] Chiara Brocchi, Brad Brown, J. M. y Neiman, M. (2016). Using agile to accelerate your data transformation.

[Chugh y Grandhi, 2013] Chugh, R. y Grandhi, S. (2013). Why business intelligence? significance of business intelligence tools and integrating bi governance with corporate governance. *International Journal of E-Entrepreneurship and Innovation (IJEEI)*, 4(2):1–14.

[D. Turk y B, 2002] D. Turk, R. F. y B (2002). Limitations of agile software processes. *British Journal of Ophthalmology*.

[Davenport y Harris, 2006] Davenport, T. H. y Harris, J. G. (2006). *Competing on Analytics: The New Science of Winning*. Harvard Business Press.

[Desai y Guy, 2019] Desai, R. y Guy, D. (2019). Ksql : Streaming sql engine for apache kafka hojjat jafarpour.

[Eckerson, 2010] Eckerson, W. W. (2010). *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. John Wiley & Sons.

[Few, 2013] Few, S. (2013). *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. Analytics Press.

[Ines y Turnbull, 2018a] Ines, B. y Turnbull, J. (2018a). Grafana: Up & running: Visualization and analytics for everyone.

[Ines y Turnbull, 2018b] Ines, B. y Turnbull, J. (2018b). Prometheus: Up & running: Infrastructure and application performance monitoring.

[Jethani *et al.*, 2021] Jethani, H., Patel, D., y Bhatt, A. (2021). Big data analytics using google bigquery. En *IOP Conference Series: Materials Science and Engineering*, volumen 1094, p. 012063. IOP Publishing.

[Joshua y Ratnam, 2018] Joshua, S. y Ratnam, K. (2018). *Agile Analytics: Applying in the Development of Data Warehouse for Business Intelligence System in Higher Education*, pp. 1038–1048.

[Kafka, ] Kafka, A. Kafka connect. `https://kafka.apache.org/documentation/#connect`. Accessed: 2022-10-12.

[Kent Beck, 2009] Kent Beck, Mike Beedle, A. v. B. A. C. W. C. M. F. J. G. J. H. A. H. R. J. J. K. B. M. R. C. M. S. M. K. S. J. S. y. D. T. (2009). Agile alliance. manifesto for agile software development.

[Kimball y Ross, 2008] Kimball, R. y Ross, M. (2008). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons.

[Kleppmann, 2017] Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*.

[Kreps *et al.*, 2011] Kreps, N., Narkhede, N., Rao, J., y et al. (2011). Apache avro: A compact, fast, binary data format. a detailed study about avro. Apache Software Foundation.

[Labs, 2021] Labs, G. (2021). Grafana: The open observability platform.

[Ladas, 2009] Ladas, C. (2009). *Scrumban: Essays on Kanban Systems for Lean Software Development*. Modus Cooperandi Press.

[Louden, 2002] Louden, K. (2002). *Monitoring: An essential practice for distributed system performance*. CRC Press.

[López, 2022] López, J. M. (2022). *Optimización y gestión avanzada en BigQuery*. Editorial de Tecnología.

[Marr, 2015] Marr, B. (2015). *Big Data: Using Smart Big Data, Analytics and Metrics to Make Better Decisions and Improve Performance*. John Wiley & Sons.

[Negash, 2004] Negash, S. (2004). Business intelligence. *The Communications of the Association for Information Systems*, 13(1):177–195.

[Ranjan, 2008] Ranjan, J. (2008). Business intelligence: Concepts, components, techniques and benefits. *Journal of Theoretical and Applied Information Technology*, 9(1):60–70.

[Santiago Comella-Dorda y Speksnijder, 2016] Santiago Comella-Dorda, S. L. y Speksnijder, G. (2016). An operating model for company-wide agile development.

[Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. pp. 336–343.

[Sutherland, 2014] Sutherland, J. (2014). *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Business.

[Turban *et al.*, 2011] Turban, E., Sharda, R., y Delen, D. (2011). *Business Intelligence: A Managerial Perspective on Analytics*. Pearson.

[Watson, 2010] Watson, H. J. (2010). Bi-based organizations. *Business Intelligence Journal*, 15(2):4–6.

[y M. Garcia, 2017] y M. Garcia, D. (2017). Hefesto data warehousing: Guía completa de aplicación teórica - práctica.

[Zikopoulos y Eaton, 2012] Zikopoulos, P. y Eaton, C. (2012). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill Osborne Media.