

Proyecto de pasantía:
“Potenciando a los Sellers: Modelo de mejora para
mejorar y medir su performance en Walmart
Marketplace”

Alumno: Felipe Alfonso Cornejo Correa

Carrera: Ingeniería Civil Informática

Profesor: Víctor Nivia

Fecha: 1 de Diciembre 2023

Índice

1	Resumen Ejecutivo	3
1.1	Resumen Ejecutivo (Español):	3
1.2	Abstract (English):	4
2	Introducción:	5
2.1	Contexto:	5
2.2	Identificación de oportunidad:	9
3	Objetivos:	10
3.1	Objetivo General:	10
3.2	Objetivos específicos:	10
4	Estado del arte y propuestas de solución:	10
4.1	Estado del arte	10
4.2	Propuestas de solución:	13
4.3	Solución escogida:	14
5	Métricas, desarrollo del proyecto y plan de implementación:	15
5.1	Medidas de desempeño:	15
5.1.1	Tasa de cancelación:	16
5.1.2	Tasa de devolución:	16
5.1.3	Medidas/Metodologías para objetivos específicos:	17
5.2	Evaluación económica:	18
6	Desarrollo del proyecto:	19
6.1	Diseño:	22
6.2	Piloto:	23
7	Resultados:	24
8	Conclusiones y discusión:	28
9	Referencias:	29
10	Anexos:	30

1 Resumen Ejecutivo

1.1 Resumen Ejecutivo (Español):

En el presente informe se resume la experiencia de pasantía del alumno en el área de Marketplace de Walmart Chile, donde se trabajó e implementó un proyecto en el equipo de Seller Performance el cual se dedica a analizar y reportar el comportamiento del seller en el negocio.

El objetivo de este proyecto consiste en mejorar la eficiencia del análisis de los sellers, mediante la implementación de un modelo de mejora que facilite el análisis/medición del comportamiento de los sellers mediante el envío de correos automáticos para así disminuir las tasas de cancelación y devolución, ya que día a día ingresan nuevos sellers a Walmart Marketplace. Para lograr esto se utilizaron distintas tecnologías como bases de datos en Google Big Query código en SQL, Python y HTML, y la implementación de una API para poder tener métricas de cómo es que se comportan los sellers a la hora de recibir el correo.

Se logró crear una nueva estrategia de comunicación para los sellers dentro de la empresa y una nueva forma de hacer un análisis y obtener métricas de los sellers. También se logró reducir en un 100% el error humano que podría existir y el tiempo en la comunicación con los sellers también se vio afectado positivamente, ya que se redujo al mínimo.

En el primer semestre del 2024 se espera internacionalizar el Marketplace (es decir, permitir que sellers chilenos puedan vender en el extranjero y sellers extranjeros puedan vender en Chile) y por ende se espera que este correo reciba cambios en el lenguaje y los umbrales de medición que posee, pero se seguirá utilizando el mismo código. También se espera seguir automatizando la comunicación con los seller utilizando el mismo sistema implementado, que contiene un API para poder tener las métricas que se pueden obtener hoy en día con este proyecto.

Palabras clave: seller, código, API, métricas

1.2 Abstract (English):

This report summarizes the student's internship experience in the Marketplace area of Walmart Chile, in which a project was developed and implemented for the Seller Performance team, which is dedicated to analyzing and reporting a seller's behavior within the business.

The objective of this project is to improve the efficiency of the seller analysis process through the implementation of an improvement model that facilitates the analysis/measurement of seller behavior by automatically sending emails in order to reduce cancellation and return rates, since new sellers enter Walmart Marketplace every day. In order to achieve this, different technologies were used such as databases in Google Big Query, codes as SQL, Python and HTML, and the implementation of an API allowing to have metrics on how sellers behave when receiving the email.

It was possible to create a new communication strategy for sellers within the company and a new way of doing an analysis and obtaining seller metrics. It was also possible to reduce the human error on a 100% that could exist and the time in communication with sellers was also positively affected, since it was reduced to a minimum.

Further on, in the first half of 2024 it is expected to internationalize the Marketplace (this will allow Chilean sellers to sell abroad and foreign sellers to sell in Chile) and, with this in mind, it is expected that this email process will receive changes in the language and measurement thresholds it includes, albeit the same code will be usable as-is. It is also expected to continue automating communication with sellers using the same system that has been implemented, this contains an API in order to have the same metrics that can be obtained today with this project.

Key Words: seller, code, API, metrics

2 Introducción:

2.1 Contexto:

Walmart es una corporación multinacional de origen estadounidense, fue fundada por Sam Walton el 2 de julio de 1962 en Arkansas, Estados Unidos. Esta empresa se autodescribe como una empresa que tiene como pilar de estrategia tener precios bajos todos los días, venden desde comida y entretenimiento hasta artículos deportivos y artesanías, ofreciendo una amplia variedad que los clientes agradecen. Existen distintos canales de venta, se puede comprar a través de Walmart.com, a través de una aplicación móvil o en una tienda. Por otra parte, Walmart a nivel internacional tiene más de 5.100 empresas minoristas y 550.000 asociados en todo el mundo, además de trabajar con fabricantes para vender sus productos en todo el mundo, manteniéndose así la reputación de Walmart como el minorista más confiable del mundo¹.

En Chile, Walmart es uno de los principales actores en la industria de las ventas por menor del país (retail). Su principal actividad consiste en venta de alimentos y mercaderías en supermercados (Líder, Express de Líder, SuperBodega aCuenta y Central Mayorista) y a través de internet que se distribuye en todo el territorio nacional desde Arica a Punta Arenas.²

En noviembre de 2022 comenzó a estar en marcha un nuevo negocio llamado Walmart Marketplace el cual permite que negocios de todo el país puedan vender sus productos de manera fácil y segura a través de su sitio web Lider.cl³.

Dentro del comercio online de Walmart (Lider.cl) tenemos tres distintas “zonas” cada una muy distinta de la otra:

1P: Consiste en las ventas que salen directamente de las bodegas de Walmart.

2P: Consiste en las ventas en verde que tiene Walmart.

3P: Consiste en las ventas que hacen empresas externas (como por ejemplo pymes) a través de Walmart, siendo este un intermediario entre vendedor y persona.

Walmart Marketplace⁴ está ubicada en la zona 3P ya que, como se mencionó anteriormente, permite que negocios de todo el país puedan vender a través de su sitio web.

¹ <https://corporate.walmart.com/about>

² <https://www.walmartchile.cl/>

³ <https://www.walmartchile.cl/walmart-chile-lanza-marketplace-integrado-a-plataforma-global-de-la-compania/>

⁴ <https://www.lider.cl/catalogo> (Link al sitio web de Líder en donde se encuentra el Marketplace de Walmart)

Por otra parte, es importante mencionar que dentro de las oficinas de Walmart Marketplace existen distintos equipos, en las que cada uno tiene un rol distinto:

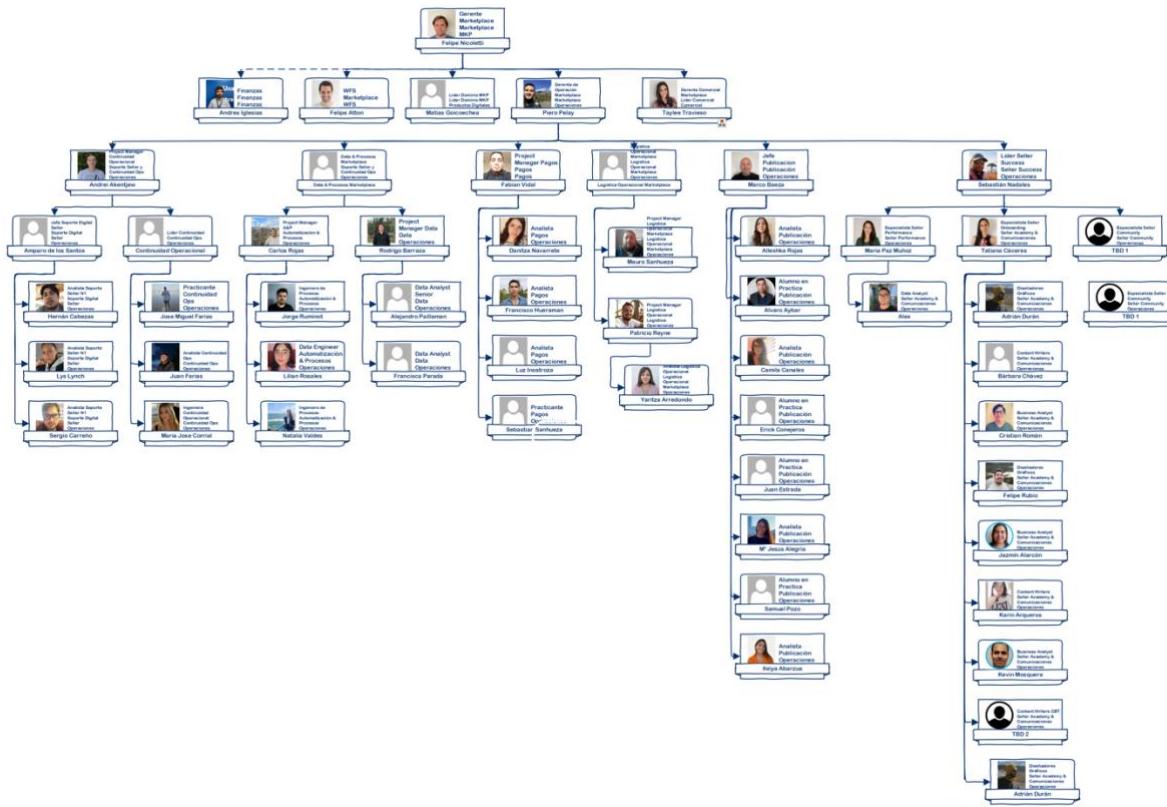


Imagen 1: Organigrama Walmart Marketplace. Fuente: Walmart Marketplace

El desarrollo del proyecto está ubicado en el equipo de Seller Success específicamente en el sub-equipo de Seller Performance (es decir, del comportamiento del vendedor), este se encarga de analizar y reportar cómo van las ventas de estos sellers (pymes/empresas que venden a través del sitio web Lider.cl), si sus ventas han sido exitosas, canceladas o devueltas y la razón de todo esto. Actualmente la única forma actual que los seller pueden saber cómo va su performance es a través de una notificación que se les envía por correo cuando avanzan de strike. Actualmente existen 4 tipos de strikes:

Strike 0: Es una advertencia luego de 1 semana con mala performance.

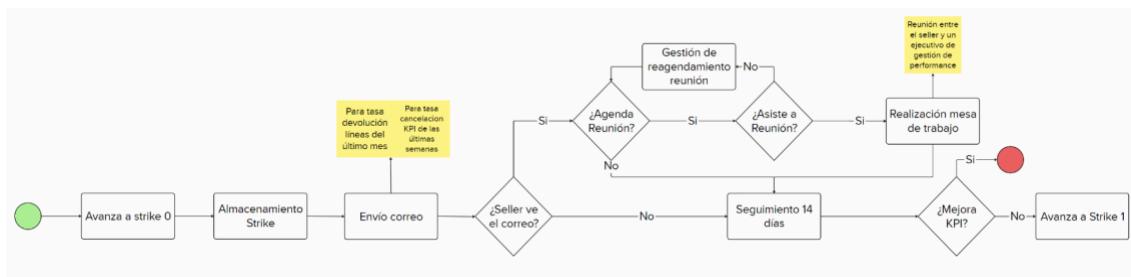


Imagen 2: Diagrama de flujo strike 0. Fuente: Elaboración propia, Walmart Marketplace.

Strike 1: Es una advertencia luego de 2 o más semanas con mala performance y el seller es citado a una reunión. (Seller critico es un seller con muy mala performance)

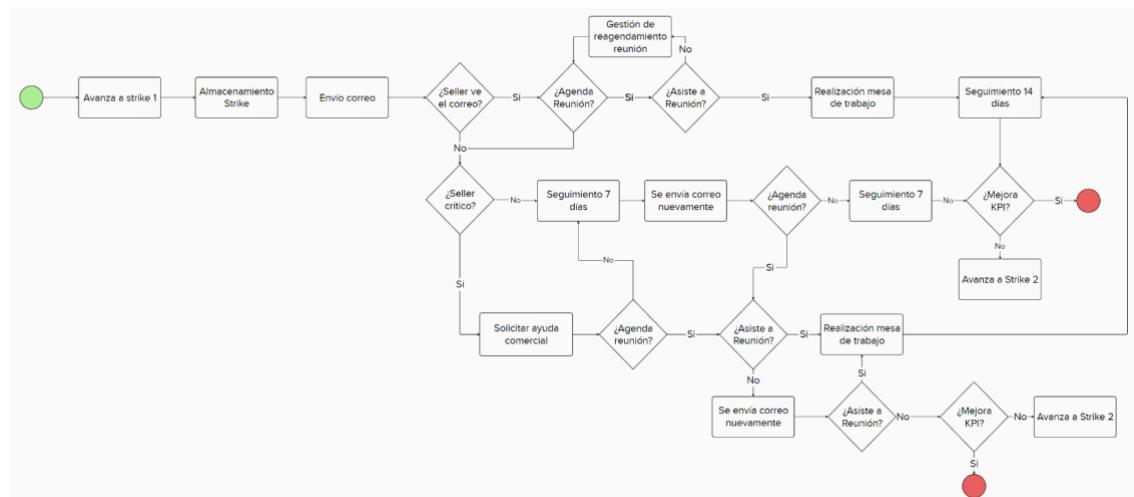


Imagen 3: Diagrama de flujo strike 1. Fuente: Elaboración propia, Walmart Marketplace.

Strike 2: Es una advertencia luego de 1 semana con mala performance luego del strike 1, y si el seller no asiste a reunión y/o tampoco hay una mejora, se suspende de la plataforma hasta que haya mejora o asista a una reunión.

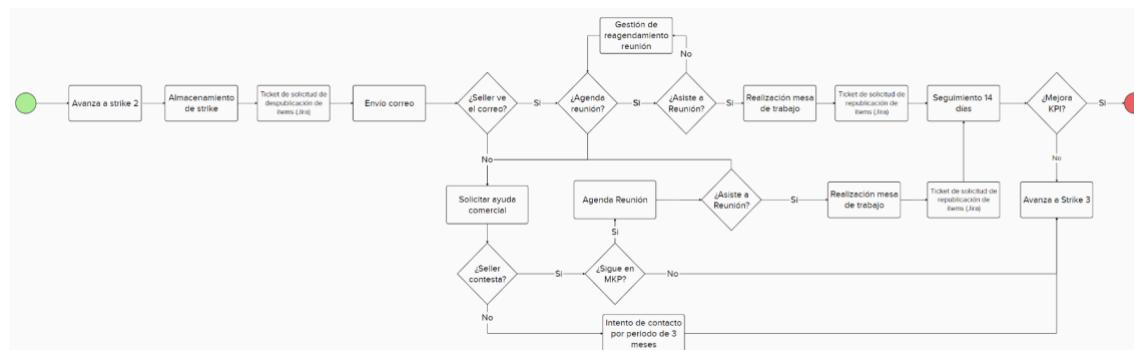


Imagen 4: Diagrama de flujo strike 2. Fuente: Elaboración propia, Walmart Marketplace.

Strike 3: El seller es dado de baja de la plataforma.

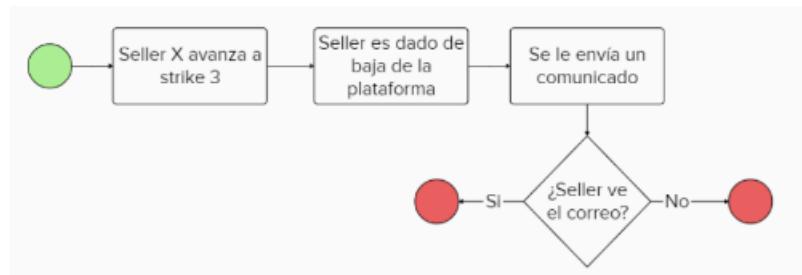


Imagen 5: Diagrama de flujo strike 3. **Fuente:** Elaboración propia, Walmart Marketplace.

Esta notificación se debe enviar manualmente aproximadamente a un mínimo de 30-35 sellers que avanzan de strike semanalmente (y puede existir el caso que una semana sean más), en esta notificación se incluye: información sobre el proceso de penalidades, un enlace para agendar una reunión, un recordatorio de que deben actualizar el estado de las órdenes y un pantallazo de su última semana de venta.

From:
To:
Subject:

Estimado:

Junto con saludar, adjunto información importante sobre sus indicadores de rendimiento Walmart Marketplace Chile.

- Adjunto informacion de proceso de penalidades <https://sellerhelp.cl.walmart.com/s/toparticles?language=es&article=guia-de-infracciones>
- Si tienes alguna duda puedes agendar una reunión en este link: <https://outlook.wal-mart.com/owa/calendar/ContactoSeller@walmart.onmicrosoft.com/bookings/vlogG/GbeUu21roHIMzdA2>
- Es muy importante destacar que debes cambiar el estado de tus órdenes, tienes 24 horas hábiles antes de que sean canceladas automáticamente, lo que podría afectar a tus indicadores de rendimiento <https://seller.walmart.com/>

Semana	Año	Partner ID	Seller	KAM	Total pedidos	Total cancelaciones	CANCELACIONES por operador	CANCELACIONES por seller	Tasa de cancelación
13	2023	[REDACTED]	[REDACTED]	[REDACTED]	43	43	0	43	100.0%

CANCELACION por Operador (cancelaciones realizadas por operador Walmart por exceder plazo de aceptación)
 CANCELACION por Seller (cancelaciones realizadas por el seller por Sin stock, error de publicación etc)

Saludos Cordiales
 Equipo Performance

Imagen 6: Correo tipo que se envía a los sellers al avanzar de strike. **Fuente:** Equipo Seller Performance, Walmart

Marketplace

Para la empresa los sellers son muy valiosos ya que, la ganancia de Walmart Marketplace es mediante las comisiones⁵ que genera cada venta y por ende esta es la única fuente de ingresos, entonces no conviene que los sellers avancen de strike. Actualmente, en la empresa no existen métricas que los sellers puedan ver fácilmente, solamente pueden ver el correo que les notifica cuando les llega un strike y ya es cada vez más común que los sellers

⁵ <https://sellerhelp.cl.walmart.com/s/toparticles?language=es&article=tabla-de-comisiones>

quieran ver más detalles de sus KPI, sus ventas o alguna forma que les permita orientarse sobre su estado actual y en caso de que avancen de strike un detalle del porqué de este, es por esto que mediante esta problemática/oportunidad se podría medir y/o prevenir estas situaciones, ya que de esta manera se podría evitar que los seller tengan un comportamiento negativo permitiéndoles visualizar su cantidad de ventas semanal/mensual, y así se podría evitar que los sellers avancen al siguiente strike.

2.2 Identificación de oportunidad:

Al ser Marketplace un área nueva es normal que los procesos sean lentos, esto puede derivarse debido a muchos factores como, por ejemplo, que hay que comprender las necesidades del cliente, optimizar procesos, los recursos son limitados, hay que adaptarse a la demanda o ajustes de estrategia. Pero en el contexto actual de la industria TI es cada vez más esencial que se pueda ahorrar tiempo y recursos, que se puedan tener los datos a mano de manera fácil, rápida y que estos sean datos concretos para una correcta toma de decisiones y una mejora de la calidad del servicio que se entrega, es por esto que en Walmart Marketplace se encontró la oportunidad para hacer una transformación digital y así crear una estrategia de comunicación, poder mejorar los tiempos de obtención de los datos, mejorar la visualización de los datos para los sellers de manera que no existan fugas de información sensible y poder hacer un análisis de datos para poder seguir mejorando el servicio entregado hacia los sellers, de tal manera que se les pueda entregar su información semanal/mensual que contenga los puntos mencionados anteriormente. Actualmente el dolor consiste en que no hay una forma periódica de comunicarse con los seller debido a la gran cantidad de tiempo que toma obtener datos y posteriormente analizarlos para enviarlos a los Sellers o implementar mejoras en el servicio, entonces como consecuencia no se pueden compartir los datos de manera rápida, eficaz, ni segura ya que, al momento de compartirlos se podría filtrar información sensible por algún error humano de la misma empresa o bien no se podrían reflejar los datos correctos si se hace de forma rápida o apurada. Por esta razón y en este contexto, se decidió comenzar el proyecto de implementar un modelo que permita agilizar los tiempos de comunicación, obtención, visualización y análisis de los datos de los sellers de manera rápida, segura y con datos precisos, para así poder aumentar las ventas, no pausar más publicaciones y poder guiar y orientar a los Sellers de la mejor manera para que tengan un comportamiento positivo.

3 Objetivos:

3.1 Objetivo General:

El objetivo general del proyecto consiste en mejorar la eficiencia de la comunicación y el análisis de los sellers, mediante la implementación de un modelo de mejora que facilite el análisis/medición del comportamiento de los sellers mediante el envío de correos automáticos para así disminuir las tasas de cancelación y devolución.

3.2 Objetivos específicos:

Los objetivos específicos del proyecto son⁶:

1. Mejorar la comunicación de KPI hacia sellers, entregándoles su información por semana/mes y un análisis de sus datos.
2. Generar un impacto económico positivo en la empresa para sellers que tengan valores negativos en sus ventas (cancelaciones y devoluciones).
3. Reducir la cantidad de sellers que avanza de strike mensualmente.
4. Reducir el tiempo que toma enviar correos manualmente con las notificaciones de strikes.
5. Aumentar el crecimiento de los sellers a través de la entrega de los KPI de sus ventas de las últimas 4 semanas (sean positivos o negativos).
6. Evitar la filtración de información por error humano mediante el envío automático de los correos a sus respectivos destinatarios con la información justa y necesaria.
7. Dejar implementado un registro dentro de la empresa que permita tener un monitoreo automático de los sellers activamente.

4 Estado del arte y propuestas de solución:

4.1 Estado del arte

Actualmente la automatización de distintos procesos de gestión en el área de informática ha aumentado significativamente debido a los beneficios que puede otorgar. Citando a Begnini Domínguez, L. F., Lecaro Lavayen, A. C., & Shauri Romero, J. D. (2022, 15 de julio). *Ventajas de la automatización de la gestión por procesos.* Recuperado de [<https://dspace.itsjapon.edu.ec/jspui/handle/123456789/3360>] “La automatización de la

⁶ Objetivo SMART en Anexo 2

gestión por procesos se puede afirmar que entre todos los objetivos que posee la reducción de costos es fundamental, alcanzando esta reducción por medio de las integraciones entre aplicaciones con el fin de reemplazar los procesos manuales. De esta forma se potencia el tiempo de atención de las actividades y tareas, además de minimizar los errores humanos susceptibles de cometerse al desarrollar las labores de forma manual”.

Históricamente la automatización de procesos en la industria ha tenido un gran impacto, el principal es que otorga una eficiencia corporativa, logrando así, un ahorro de tiempo y recursos. Pero también tiene un lado negativo, en la literatura la automatización de procesos ha sido objeto de un análisis exhaustivo, revelando una dualidad de percepciones entre sus impactos positivos y negativos, citando a García Ramírez, E. G. (2020). Impactos de la automatización en procesos laborales, percibidos por un grupo de líderes de gestión humana de una organización del sector manufacturero de la ciudad de Medellín. Recuperado de https://repository.eafit.edu.co/bitstream/handle/10784/16065/EdwinGiovany_GarciaRamirez_2020.pdf?sequence=2&isAllowed=y “*La historia está llena de ejemplos de tecnologías que aniquilaron industrias enteras, pero que al mismo tiempo crearon otras industrias que generaron muchos más empleos. De cualquier manera, no es nada seguro que en el futuro ocurra lo mismo, pues la automatización de los trabajos, los avances de la inteligencia artificial y la aceleración tecnológica son cada vez mayores.”*

Pero analizando las ventajas de la automatización a nivel industrial, la cámara de comercio de Medellín dice lo siguiente: “*Disminuye los costos de fabricación; aumenta la eficiencia del proceso productivo; agiliza la respuesta a las demandas del mercado; (...) favorece la competitividad empresarial; mejora la seguridad de los procesos y la calidad del producto; permite hacer un control de calidad más exacto*” (Cámara de Comercio de Medellín, 2019). Impactos de la automatización en procesos laborales, percibidos por un grupo de líderes de gestión humana de una organización del sector manufacturero de la ciudad de Medellín. Recuperado de https://repository.eafit.edu.co/bitstream/handle/10784/16065/EdwinGiovany_GarciaRamirez_2020.pdf?sequence=2&isAllowed=y

En el contexto del proyecto, para lograr este tipo de automatizaciones, en la industria actualmente es muy común utilizar servicios de “Email Marketing”, ya que esto facilita el envío y creación de correos, permite programar el envío de correos desde el mismo sitio web y te otorga un servicio para crear plantillas. Dentro de este servicio con un par de clicks se pueden hacer correos muy completos sin necesidad de programar, otorgando así un servicio no code⁷. Para esto las empresas actualmente a nivel mundial (incluido Chile) utilizan servicios como: Mailchimp, Zoho Campaigns, Drip, MailerLite, Brevo, Campaigner, GetResponse, Moosend, ActiveCampaign y ConstantContact⁸, de esta manera contratando un servicio de pago mensual, pueden organizar campañas de envío de correos masivos para así llegar a sus clientes de manera fácil, rápida y eficaz.

Y en Walmart no es diferente, ya que en otras gerencias se utiliza el servicio de Salesforce⁹, el cuál es un servicio de CRM que otorga muchos software basados en la nube y dentro de esos servicios podemos encontrar el servicio de Salesforce Marketing Cloud¹⁰.

Para el caso de este proyecto, se enfocará en el desarrollo de un proceso de automatización para el envío, gestión y medición de correos electrónicos¹¹ al tener una condición activa (para los sellers que tengan al menos 1 venta en las últimas 4 semanas), para mejorar la eficiencia, seguridad de los datos, desarrollo de los sellers y así lograr una mejora en la comunicación hacia los sellers y obtener beneficio de esto. Posteriormente, con el envío de estos correos se buscará tener un impacto en la tasa de cancelación/devolución para poder hacer un análisis y medir si es que el proyecto tuvo o no un impacto en la empresa.

En la empresa se desea que este proyecto otorgue un beneficio positivo, ya que, se busca crear una estrategia de comunicación con los sellers, reducir al mínimo el tiempo en el envío de estos correos, también se busca tener un impacto en reducir el error humano, ya que, en tareas repetitivas es muy común que exista, y por ende así se tendría un impacto positivo en la calidad y precisión del envío del contenido del correo.

⁷ Plataformas para usuarios que no tienen conocimiento en programación, permite crear software sin código.

⁸ <https://www.forbes.com/advisor/business/software/best-email-marketing-software/>

⁹ <https://www.salesforce.com/>

¹⁰ <https://www.salesforce.com/es/products/marketing-cloud/overview/>

¹¹ Se medirá la Tasa de apertura, Tasa de rebote y Tasa de clicks.

4.2 Propuestas de solución:

Es importante mencionar que todas estas propuestas de solución serán mediante el envío de correos automáticos, pero la gran diferencia será en la forma en que este correo entregará los datos y la facilidad que tendrá la empresa posteriormente para hacer un análisis y una medición de esto.

Correo con botón con enlace a un sitio web para que los sellers vean sus KPI:	La propuesta consiste en crear dashboards en la nube para que los sellers revisaran sus métricas de desempeño a través de un enlace único por cada seller en un correo. Sin embargo, fue descartado debido a riesgos de seguridad de datos, por que implicaba extraer información confidencial de la empresa hacia un host que permita acceso a externos, entonces existe posibilidad de filtración de datos y por políticas restrictivas de la empresa y la incertidumbre sobre la entrega efectiva del correo a los vendedores fue descartada.
Uso de una máquina virtual para envío de correos:	La propuesta consistía en ejecutar un código semanal en una máquina virtual para enviar correos electrónicos con tablas personalizadas de cada seller extraídas desde Google de Big Query. Sin embargo, fue descartada porque no proporcionaba métricas de seguimiento, lo que limitaba la capacidad de evaluar la efectividad de la estrategia de comunicación en términos cuantitativos y cualitativos.
Envío de correos incluyendo una API en el código para monitorear métricas:	La propuesta, es casi igual a la anterior, se diferencia en que se añade el uso de una API de servicio de “email marketing” para enviar los correos. Esto permite medir métricas clave, como la tasa de apertura ¹² , tasa de rebote ¹³ y tasa de clicks ¹⁴ , proporcionando métricas detalladas sobre el rendimiento. Además, los resultados serían accesibles internamente en la nube a través del sitio web del servicio, facilitando información sobre la interacción de los receptores y mejorando la estrategia de comunicación.

Tabla 1: Propuestas de solución. **Fuente:** Elaboración propia

¹² Indicador para saber cuántas personas abrieron y vieron el correo.

¹³ Indicador para saber cuántos correos no llegaron a su destinatario por que el correo no existe o está mal escrito.

¹⁴ Indicador para saber cuántas interacciones/reuniones se hicieron/agendaron mediante los correos.

4.3 Solución escogida:

Analizando detalladamente y pensando en el futuro de la empresa se decidió que la solución escogida sería la de envío de correos incluyendo una API en el código para monitorear métricas, ya que así se podrá ver y comprobar si la solución es realmente efectiva. Pero para fundamentar es importante mencionar los puntos del proceso de solución y por qué se escogió esta solución¹⁵:

1. Para seleccionar los datos y la tabla que se incluirá en el correo: Se trabajará desde Google Big Query, que es donde se encuentran las bases de datos de Walmart Marketplace que trabajan a prácticamente a tiempo real, de aquí se puede obtener una tabla con los KPI de todos los seller, posteriormente esta tabla se llevará y filtrará para cada seller en Python, luego ya teniendo la tabla filtrada se le enviará a cada seller con su información cuando se cumpla la condición¹⁶ y así se enviaría la tabla con la última actualización. (Anexo 3). En la tabla se planea incluir las ventas totales, cancelaciones por seller y cancelaciones hechas por Walmart que son métricas que existen en la base de datos, además se planea incluir la tasa de cancelación y en otro correo enviar la tasa de devolución
2. Para automatizar el envío de correos: Se creará un código en Python integrando la API de Salesforce. Se utilizarán librerías para extraer los datos desde Google Big Query hacia Python¹⁷, luego se automatizará el proceso de ejecución del código utilizando la plataforma “element”¹⁸ de Walmart, la cual nos permite ejecutar el código cada cierto periodo determinado (en este caso se ejecutará cada 7 días para código de cancelaciones y 1 mes para devolución).
3. Integrar una API: Específicamente utilizar un servicio de “email marketing” de Salesforce que tenga la opción de integrar API al código para enviar correos, dado que esto permite

¹⁵ Matriz de riesgos en Anexo 6

¹⁶ Tener al menos 1 venta en las últimas 4 semanas

¹⁷ <https://cloud.google.com/python/docs/reference/bigquery/latest>

¹⁸ <https://medium.com/walmartglobaltech/walmarts-multi-cloud-machine-learning-platform-a1ab08ff1e4a>

tener un mayor control de lo que está pasando con el envío de los correos, ya que permite obtener las distintas métricas¹⁹. Esta API se integrará mediante código a nuestro proyecto, y como se dijo anteriormente de esta manera podemos tener métricas posteriores al envío de estos correos para ver concretamente quién no está interactuando con el correo para poder actuar de manera más específica.

4. Pensando en el futuro de la empresa: Utilizar una API es esencial, ya que nos facilita la comunicación entre aplicaciones/plataformas²⁰ y actualmente el uso de APIs brinda flexibilidad, eficiencia y la capacidad de aprovechar oportunidades emergentes, lo que es esencial para el crecimiento y la competitividad de una empresa.

5 Métricas, desarrollo del proyecto y plan de implementación:

A continuación se detallará todo el proceso de desarrollo del proyecto:

5.1 Medidas de desempeño:

La medición de los resultados del proyecto al implementarse será utilizando distintas métricas. La primera métrica que se medirá será el impacto que tenga en los sellers el envío de sus KPI, y esto se podrá medir revisando si la cantidad de sellers que avanzan de strike disminuye o aumenta, ya que semanalmente se estima que entre 3% a 5% de los sellers está avanzando de strike semanalmente, esto se medirá con la tasa de cancelación y la tasa de devolución.

También se puede medir visualizando si posterior al envío de los KPI, las tasas de cancelación/devolución de los sellers bajan. Todo esto se puede visualizar ya actualmente, debido a que existen dashboard de información en la empresa para esto:

¹⁹ Tasa de apertura, Tasa de rebote y Tasa de clicks.

²⁰ <https://www.forbesargentina.com/innovacion/api-futuro-software-y-economia-n3856>

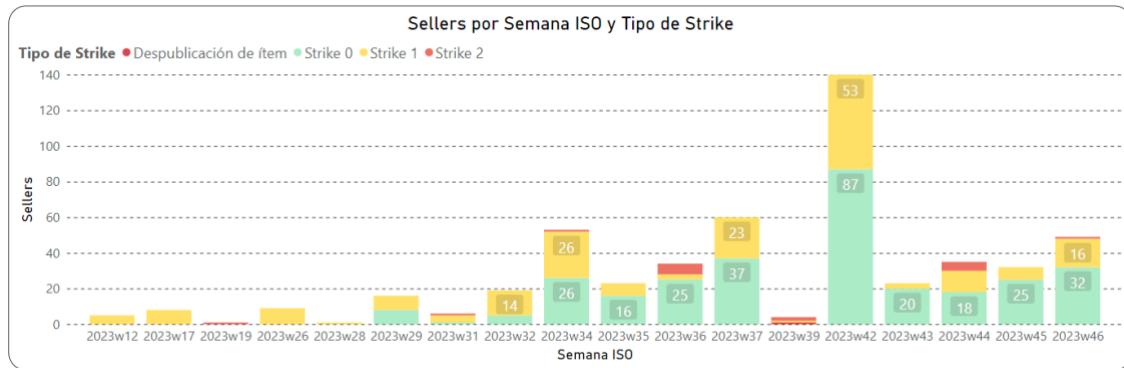


Imagen 7: Dashboard de sellers en strike. Fuente: Walmart Marketplace, Seller Performance.

5.1.1 Tasa de cancelación²¹:

$$\frac{(CANCELACIONES TOTALES POR SELLER + CANCELACIONES TOTALES POR OPERADOR) * 100}{TOTAL DE ÓRDENES}$$

CANCELACIONES TOTALES SELLER: Son todas las cancelaciones hechas por el mismo seller.

CANCELACIONES TOTALES OPERADOR: Son las cancelaciones hechas por Walmart o el Courier.

Si su tasa de cancelación de ventas (independientemente sea por operador o por ellos mismos por eso se toma el total) es mayor a 6% se clasifica como un mal comportamiento, si está entre 4% y 5.9% es un comportamiento regular y si este es menor a 4% es un buen comportamiento.

5.1.2 Tasa de devolución²²:

$$\frac{(DEVOLUCIONES TOTALES) * 100}{TOTAL DE ÓRDENES}$$

DEVOLUCIONES TOTALES: Son todas las devoluciones hechas por los clientes por motivos variados, ya sea por garantía o por algún derecho legal del consumidor.

²¹ Como se extrae esta información en el código, ver anexo 3

²² Como se extrae esta información en el código, ver anexo 4

Si su tasa de devolución de ventas es mayor a 15% se clasifica como un mal comportamiento, si está entre 5.1% y 15% es un comportamiento regular y si este es menor a 5% es un buen comportamiento.

Es importante tener en cuenta que para obtener estos KPI se debe hacer un filtro desde la BBDD en que se encuentran, esta información está organizada en varias tablas. Las tablas principales son: OMS_STERLING_PURCHASE_ORDER, OMS_STERLING_SALES_ORDER y OMS_STERLING_RETURN_ORDER (Anexo 1), es con estas tablas que se obtiene la información para medir la cantidad de cancelaciones y devoluciones de cada seller.

Todas estas tablas se trabajan a través de Google BigQuery utilizando el lenguaje de programación SQL²³, entonces para obtener los datos necesarios por cada seller debemos filtrar la información que contienen estas tablas, ya que como las tablas contienen información de todas las ventas y estados de venta de todos los seller estas crecen día a día, debido a que cada día hay nuevas ventas generadas a través de la plataforma. Y es que a través de estas ventas que podemos obtener los KPI de ventas exitosas, cancelaciones por operador, cancelaciones por seller y devoluciones.

5.1.3 Medidas/Metodologías para objetivos específicos:

- Para medir el primer objetivo específico se utilizará lo mencionado anteriormente, al seller se le enviarán sus KPIs extraídos desde Google BigQuery y estos serán mostrados mediante una tabla simplificada y con nombres de columnas que estén en un lenguaje menos técnico y más tradicional, también se le mostrará un análisis de su comportamiento semanal/mensual para indicarles si van por buen o mal camino.

- Para el segundo y tercer objetivo, se medirán los datos de los sellers que tengan un mal comportamiento de ventas y luego de implementado el proyecto se medirá si existió un cambio positivo o negativo en su comportamiento. Esto se medirá a través de la tasa de cancelación/devolución a nivel general de todos los sellers.

²³ <https://aws.amazon.com/es/what-is/sql/>

- Para el cuarto objetivo se hará una automatización del envío de estos correos cuando se cumpla una condición²⁴, al cumplirse esta condición se enviarán los correos con las notificaciones a cada seller que la cumpla, minimizando el tiempo del envío de estos.
- Para el quinto objetivo, se espera que ya habiendo implementado el proyecto y el seller teniendo toda su información y el análisis de sus KPI visible pueda darse cuenta de que está cometiendo errores, y así comenzar a planificar estrategias y/o implementar cambios para reducir la cantidad de cancelaciones y devoluciones. Para esto Walmart ofrece la opción de agendar una reunión para orientarlos/ayudarlos con este proceso.
- Para el sexto objetivo se automatizará el envío de correos.
- Para el séptimo objetivo se planea implementar en alguna plataforma de las que tenga acceso Walmart un sistema de “email marketing” y así poder tener métricas del correo.

5.2 Evaluación económica:

Este proyecto se creó utilizando los servicios ya disponibles de Walmart, por lo que no hubo inversión para aplicaciones o servicios extras de por medio. Pero, dado que al no existir una inversión económica y este proyecto es un arreglo del sistema no existe VAN y TIR (o sea, es 0). Pero si podemos hacer una evaluación/estimación con supuestos. (*Debido a un contrato de confidencialidad firmado por el alumno, no se mostrarán datos reales en las imágenes, sino un dato referencial, Anexo 7*)

	Mes					
	1	2	3	4	5	6
VALOR GENERAL DE TODOS LOS SELLER						
Sellers Activos (SA)	800	800	800	800	800	800
Ordenes de compra (OC)	44000	44000	44000	44000	44000	44000
Valor de ticket promedio (TP)	\$ 67.000	\$ 67.000	\$ 67.000	\$ 67.000	\$ 67.000	\$ 67.000
VALOR INDIVIDUAL POR SELLER						
OC promedio por seller (OC/SA)	55	55	55	55	55	55
1% de OC promedio por seller	0,55	0,55	0,55	0,55	0,55	0,55
VENTA						
Valor de TP por OC promedio de 1 seller	\$ 3.685.000	\$ 3.685.000	\$ 3.685.000	\$ 3.685.000	\$ 3.685.000	\$ 3.685.000
Valor de 1% TP por OC promedio de 1 seller	\$ 36.850	\$ 36.850	\$ 36.850	\$ 36.850	\$ 36.850	\$ 36.850
Valor de 1% TP por OC promedio de TODOS los seller	\$ 29.480.000	\$ 29.480.000	\$ 29.480.000	\$ 29.480.000	\$ 29.480.000	\$ 29.480.000
TOTAL	\$ 176.880.000					

Imagen 8: Evaluación/Estimación económica. Fuente: Elaboración propia.

²⁴ Tener al menos 1 venta en las últimas 4 semanas.

Para un periodo de 6 meses, podemos estimar que reduciendo en un 1% la tasa de cancelación de todos los seller se puede ganar un valor aproximado mensual de \$29.480.000, esto se calcula tomando la cantidad promedio total de sellers activos actualmente en la plataforma, el promedio de las órdenes de compra en un mes y el costo del ticket promedio en Walmart. El cálculo se hizo obteniendo cuánto cuesta un 1% del valor del ticket promedio de una orden de compra por seller, y al multiplicar ese valor obtenido por el total de sellers activos llegamos a que si se logra reducir en al menos 1% la tasa en un periodo de 6 meses se podría ahorrar aproximadamente \$176.880.000.

6 Desarrollo del proyecto:

Se creó un diagrama de cómo funcionarían los procesos del proyecto para tener una base de cómo es que se desarrollará el proyecto y poder hacer un mejor análisis a la hora de empezar a crearlo. También se creó un diagrama de casos de uso del proyecto implementado para tener una mejor comprensión.

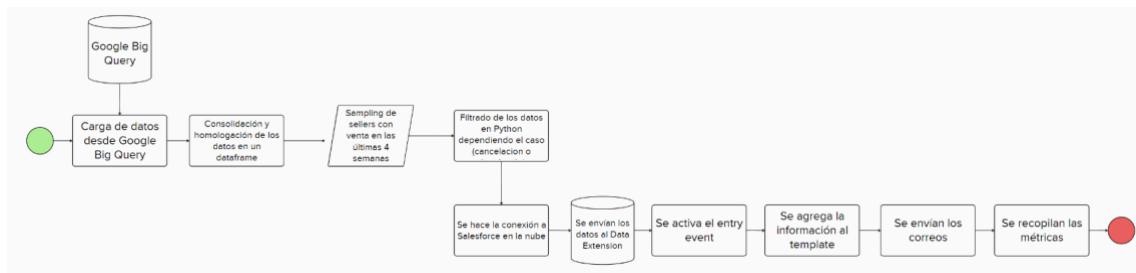


Imagen 9: Diagrama de flujo de procesos del proyecto. **Fuente:** Elaboración propia, Walmart Marketplace.

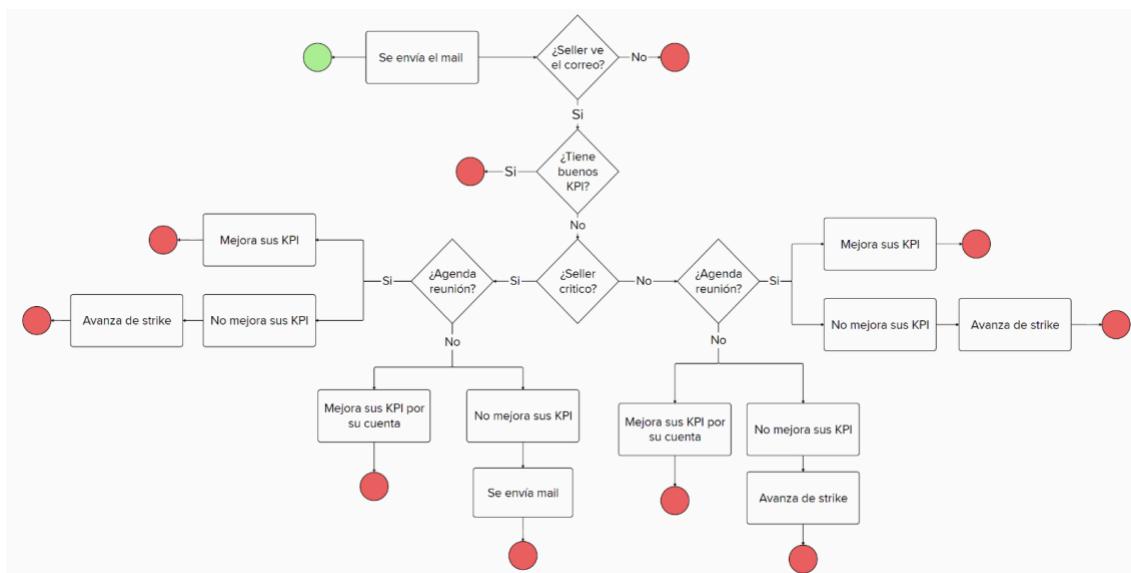


Imagen 10: Diagrama de casos de uso del proyecto. **Fuente:** Elaboración propia, Walmart Marketplace.

Para comenzar en Google Big Query se crearon 3 tablas (tabla de cancelaciones, devoluciones y una tabla que nos otorga los datos de sellers que tienen ventas en las últimas 4 semanas), posteriormente se desarrollaron 2 códigos (para cancelaciones Anexo 9 y devoluciones Anexo 10) y se llevaron estas tablas a un dataframe (tabla multidimensional) de pandas²⁵ en Python para poder obtener la información de los respectivos correos, estas tablas se modificaron dentro de Python ya que así se obtiene una mejora en el tiempo de ejecución del código y se pueden hacer operaciones más complejas y compatibles con el entorno del código.

Luego de obtener las tablas a utilizar se cambiaron los nombres de las columnas para que estas fueran más amigables al seller, ya que internamente en la empresa se usan nombres más técnicos.

Semana	Año	PARTNER_ID	SELLER_NAME	Líneas totales	Líneas canceladas por Walmart	Líneas canceladas por ti	Tasa de cancelación (%)
46	2023		CL	25	0	0	0%
45	2023		CL	23	0	0	0%
44	2023		CL	12	0	0	0%
43	2023		CL	14	0	0	0%

Imagen 11: Tabla con nombres ya cambiados. **Fuente:** Elaboración propia, Walmart Marketplace.

Para finalizar con las tablas también se utilizó la librería pretty_html_table²⁶ para darles un formato HTML a la tabla y tener así una apariencia más amigable y colorida hacia los sellers, ya que así se les puede dar colores personalizados haciéndoles ajustes respectivos y también se puede obtener el código de las tablas en HTML para insertarla en la plantilla del correo.

Por otra parte, para el envío de correos se implementó la API de Salesforce Marketing Cloud al código, de esta manera al activar el “Entry Event”²⁷ (que es cuando comienza a actuar el flujo de envío de correos) que luego de almacenar los datos en el Data Extension (Base de datos dentro de Salesforce) se envía el correo, y posterior a esto podemos comenzar a visualizar las métricas de los correos en la plataforma en la nube a medida que los sellers vayan interactuando con el correo.

²⁵ <https://pypi.org/project/pandas/>

²⁶ <https://pypi.org/project/pretty-html-table/>

²⁷ https://help.salesforce.com/s/articleView?id=sf.mc_jb_entry_sources.htm&type=5

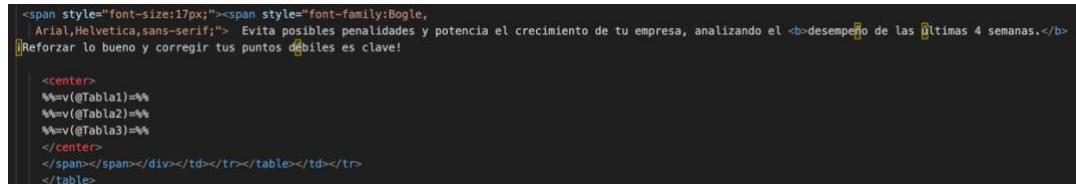
Y ya para finalizar se creó una plantilla HTML, para la cual se tomó una plantilla bases desde Salesforce y se modificó el código, ya que se agregaron atributos que no estaban en la opción “No code” que ofrece la plataforma (como por ejemplo una tabla a modo de recordatorio de los umbrales de las tasas). A esta plantilla se le deben agregar los campos variables para insertar datos desde el Data Extension de Salesforce al correo, posterior a eso se debe subir la plantilla a Salesforce en donde los nombres de los campos variables deben coincidir con los nombres de las columnas del Data Extension. En el código HTML, se deben escribir al inicio para declararlas y luego en el lugar visible que se quiere poner la variable se escribe el campo variable con el formato “`%%=v(@Variable)=%%`”.

```

1   %%[
2
3   set @Seller = Seller
4   set @Tabla1 = Tabla1
5   set @Tabla2 = Tabla2
6   set @Tabla3 = Tabla3
7
8 ]%%

```

Imagen 12: Declaración de variables al inicio del código de plantilla correos. **Fuente:** Código en lenguaje HTML de los correos.



```

Evita posibles penalidades y potencia el crecimiento de tu empresa, analizando el desempeño de las últimas 4 semanas. ¡Reforzar lo bueno y corregir tus puntos débiles es clave!
<center>
%%=v(@Tabla1)=%%
%%=v(@Tabla2)=%%
%%=v(@Tabla3)=%%
</center>
</span></span></div></td></tr></table></td></tr>
</table>

```

Imagen 13: Cómo se ven las variables en el código. **Fuente:** Código en lenguaje HTML de los correos.



```

%%[ set @Seller = Seller set @Tabla1 = Tabla1 set @Tabla2 = Tabla2 set @Tabla3 = Tabla3 ]%%

```

Walmart logo

¡Hola

%%=v(@Seller)=%%!

Estos son tus indicadores de desempeño en Walmart de tus últimas 4 semanas activas hasta hoy:

Evita posibles penalidades y potencia el crecimiento de tu empresa, analizando el **desempeño de las últimas 4 semanas**. ¡Reforzar lo bueno y corregir tus puntos débiles es clave!

%%=v(@Tabla1)=%% %%=v(@Tabla2)=%% %%=v(@Tabla3)=%%

Imagen 14: Como se ven las variables en plantilla HTML. **Fuente:** Código en lenguaje HTML de los correos.

6.1 Diseño:

Para la tasa de cancelación y devolución el diseño es casi el mismo, solo cambia un poco el texto para la tabla que corresponde y los umbrales²⁸.

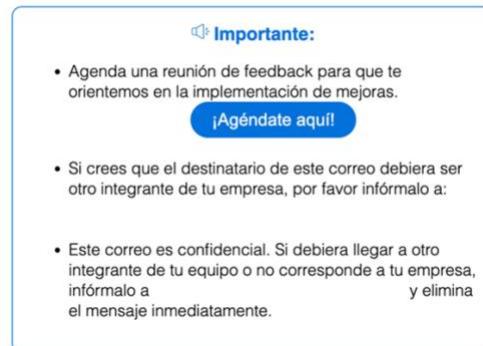
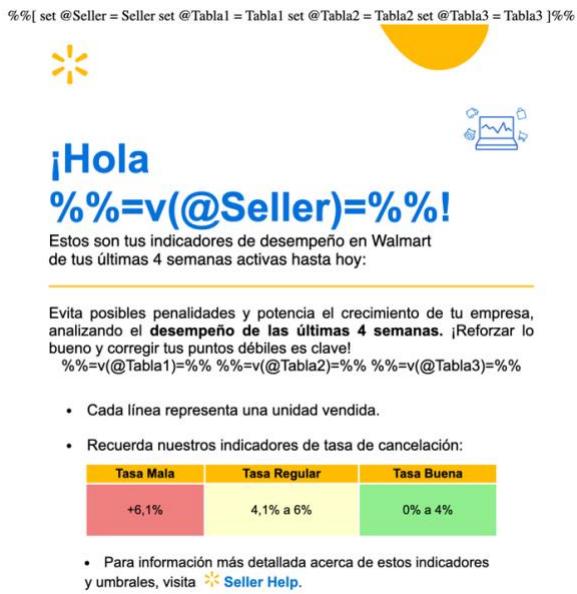


Imagen 15 y 16: Como se ve el correo antes de ingresar datos. **Fuente:** Código en lenguaje HTML de los correos.

²⁸ Por temas de espacio y organización del documento la imagen está cortada a la mitad, la imagen de la izquierda es la parte de arriba y la de la derecha la que va abajo.

6.2 Piloto:

Acá se muestra como se ve el correo enviado para los 2 distintos casos de correos²⁹.



Imagen 17 y 18: Correo que se envía a seller con tasa de cancelación. **Fuente:** Elaboración propia, Walmart Marketplace

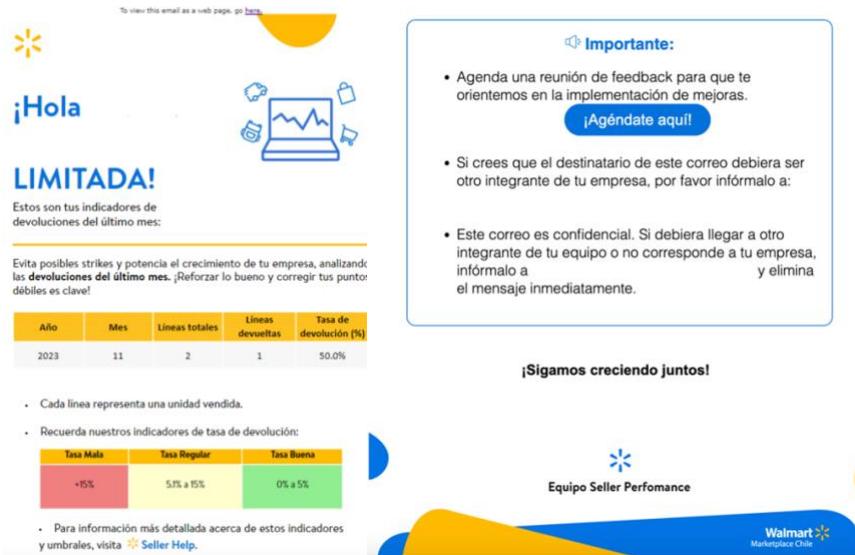
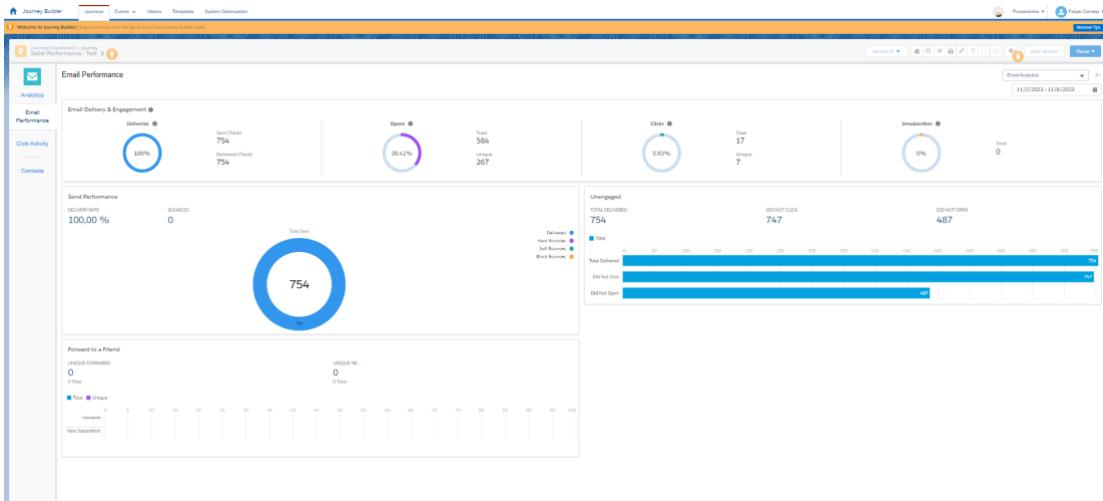


Imagen 19 y 20: Correo que se envía a seller con tasa de devolución **Fuente:** Elaboración propia, Walmart Marketplace

²⁹ Por temas de espacio y organización del documento la imagen está cortada a la mitad, la imagen de la izquierda es la parte de arriba y la de la derecha la que va abajo.

7 Resultados:

Los resultados a la fecha son positivos, ya que semanalmente podremos ver cómo es que va afectando el proyecto en el negocio, pero si podemos afirmar que se logró desarrollar e implementar el proyecto en su totalidad, creando así una estrategia de comunicación para los sellers y una nueva forma de hacer un análisis y obtener métricas de los sellers. (*Debido al contrato de confidencialidad firmado por el alumno, no se mostrarán datos en las imágenes, sino una foto de como se ve el dashboard con los datos de las métricas obtenidos al realizar pruebas internas, Anexo 8*).



Welcome to Journey Builder! Drag your mouse over the tips to learn how Journey Builder works.

Click Activity

Link	Total Clicks	Unique Clicks	Click-Through Rate
Agendate aquí	16	6	0,8 %
View as a Web Page	1	1	0,13 %
https://sellerhelp.cl.walmart.com/s/?language=es&utm_source=sfmc&utm_term=&utm_content=9641116&utm_id=60d89f39...	0	0	0 %
https://outlook.wal-mart.com/owa/calendar/	0	0	0 %
http://www.exacttarget.com/privacy-policy?l	0	0	0 %
https://view.mail.walmartdigital.cl/?j=fe92	0	0	0 %
Powered By	0	0	0 %
Seller Help	0	0	0 %
View Privacy Policy	0	0	0 %

Welcome to Journey Builder! Drag your mouse over the tips to learn how Journey Builder works.

Contacts

Contact Key	Email Address	Date
1	@walmart.com	11/8/2023 11:58:51 PM
1	@walmart.com	11/8/2023 11:58:51 PM
1	@walmart.com	11/8/2023 11:58:51 PM
1	@walmart.com	11/8/2023 11:58:51 PM
1	@walmart.com	11/8/2023 11:58:51 PM
1	@walmart.com	11/8/2023 11:58:51 PM
1	@walmart.com	11/8/2023 11:58:29 PM
1	@walmart.com	11/8/2023 11:58:29 PM
1	@walmart.com	11/8/2023 11:58:29 PM
1	@walmart.com	11/8/2023 11:58:29 PM
1	@walmart.com	11/8/2023 11:58:29 PM
1	@walmart.com	11/8/2023 11:58:29 PM

The screenshot shows the Salesforce Marketing Cloud Journey Builder interface. On the left, there's a sidebar with navigation links: 'Journey Builder' (selected), 'Journeys', 'Events', 'History', 'Templates', 'System Optimization', 'Analytics' (selected), 'Email Performance', and 'Click Activity'. The main area is titled 'Seller Performance - Test' and shows a 'Contacts' list. The list has columns for 'Contact Key' (containing values 1 through 15), 'Email Address' (all entries are '@walmart.com'), and 'Date' (timestamps from 11/8/2023 11:42:59 PM to 11/8/2023 4:14:50 PM). At the top of the main area, there are tabs for 'Sends', 'Unique Opens' (selected), 'Unique Clicks', 'Bounces', and 'Unsubscribes'. A search bar at the top right says 'Find Contact Key' with the value '10/10/2023 - 11/8/2023'.

Imagen 22, 23, 24 y 25 : Dashboards de métricas en Salesforce Marketing Cloud. **Fuente:** Walmart Marketplace

Se desarrolló el código en Python, siguiendo lo planteado en la sección del por qué se escogió esta solución, se logró conectar la BBDD desde Google Big Query a Python para obtención de datos de los sellers y posteriormente la creación de las tablas que se les enviaría a cada uno de estos. Por otra parte, se creó una plantilla en código HTML en la que se personalizó a necesidad para lograr darle una apariencia amigable para el seller y lograr que los campos fueran variables para que se rellenan de manera automática al ejecutar el código. Y finalmente la conexión del código hacia la API funciona correctamente, ya que nos permite enviar los correos a través de esta y poder ver las métricas que nos proporciona el envío a través de la API.

Respecto al cumplimiento de los objetivos el primero si fue cumplido, ya que actualmente se les está entregando su información semanal/mensual y se pueden hacer análisis con la información.

El segundo, tercero, cuarto y quinto objetivo podemos decir que van de la mano, ya que son objetivos que se pueden medir más al largo plazo, y como el proyecto lleva poco tiempo de implementación oficial, no podemos hacer un análisis/medición real del impacto muy concreto aún. Pero actualmente en el mes de diciembre de 2023 las métricas se ven así:

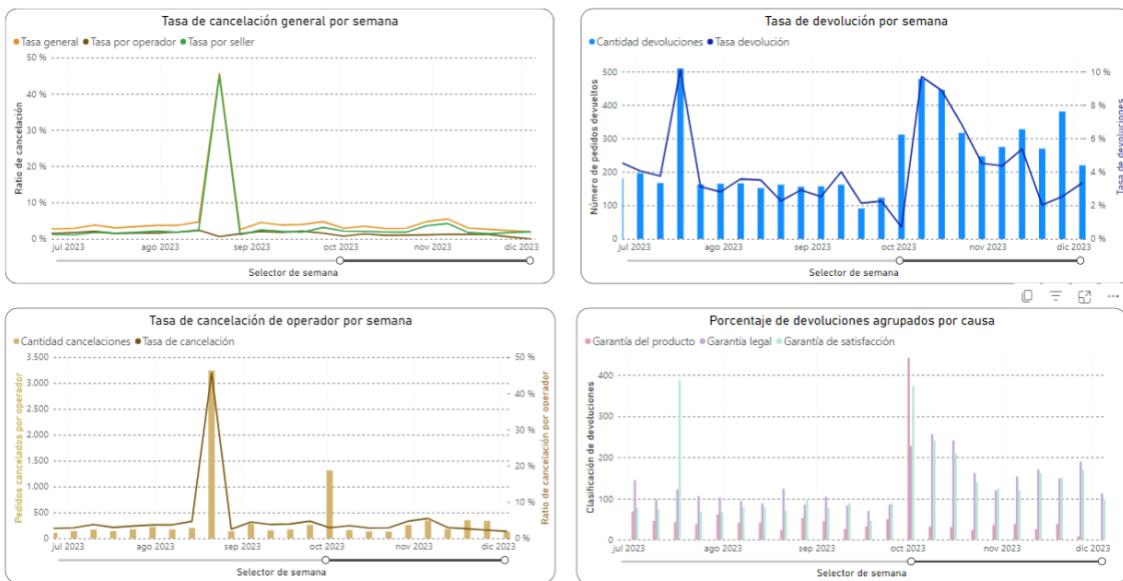


Imagen 26: Dashboard de métricas para tasas en Walmart Marketplace. **Fuente:** Walmart Marketplace

En el gráfico anterior se pueden ver las métricas actualizadas a la primera semana de diciembre, en general los niveles promedio de cancelaciones por semana es entre 300-400 órdenes y de devoluciones 150-200. Como el proyecto fue implementado en la segunda semana de noviembre aún no podemos ver resultados muy notorios, más aún después de una fecha de compras masivas como lo fue para el caso del Black Friday 2023 (desde el 24/11 hasta el 03/11 en el caso de Walmart)³⁰. Pero, sí es posible ver que en el mes de diciembre los niveles de las tasas en ambos casos comienzan a bajar, por lo que podría relacionarse al proyecto. Entonces sí podría estar teniendo un impacto ya que, al haber menos cancelaciones se puede asociar a una mejoría en las comisiones, también a menos sellers que avanzan de strike debido a la disminución de las tasas por cancelaciones por seller y por ende podría existir un crecimiento en las ventas de los sellers.

El sexto objetivo también se logró cumplir, ya que al haber hecho la automatización de este proceso se evita la intervención humana en el envío y configuración del correo, reduciendo de esta manera el error humano en un 100% en este proceso. Y por ende el séptimo objetivo también fue cumplido, ya que este proyecto quedó implementado en la empresa y permite tener un monitoreo automático de estos, ya que no hay procesos manuales, solo el tener que ingresar manualmente al sitio web de Salesforce Marketing Cloud para visualizar el reporte con

³⁰ <https://www.vogue.mx/articulo/black-friday>

las mediciones/información en los dashboard que nos otorga la plataforma que se actualiza a tiempo real.

8 Conclusiones y discusión:

Durante la pasantía en Walmart surgió la oportunidad de desarrollar un proyecto, el proyecto permitió aplicar y expandir conocimientos universitarios, y también permitió aprender nuevos conocimientos en el ámbito informático para el desarrollo en el futuro.

En el período de la pasantía, se logró implementar el proyecto exitosamente con todo lo que se propuso en un comienzo, en el que se incluye el envío automático de los correos y un dashboard en la nube que permite ver las métricas de la información que se obtiene a partir de esto.

Esta experiencia permitió al alumno aprender a trabajar en profundidad con tecnologías como SQL, Python, HTML, Salesforce y Salesforce Marketing Cloud (y APIs), lo que le permitió adquirir habilidades para la automatización de procesos y obtención de información a partir de esto. Además, generó instancias para mejorar la capacidad de abordar desafíos complejos de programación y para adaptarse a nuevos entornos de desarrollo (en este caso el estricto y privado entorno de Walmart).

El alumno de pasantía dentro de la empresa fue incluido como si fuera uno más del equipo, se le incluyó en reuniones diarias, en las planificaciones y toda opinión era importante al momento de tomar decisiones porque era incluido en todo lo que se podía, lo que aportó al éxito general del proyecto, ya que le permitió conocer en primera persona como es que funciona un equipo de trabajo en un entorno de una empresa real.

También, como se mencionó anteriormente, se tuvo que enfrentar a nuevos desafíos los cuales fueron integrar una API a un código en Python y ejecutar consultas SQL a grandes conjuntos de datos. El enfrentar estos desafíos brindaron la oportunidad para mejorar las habilidades frente a la resolución de problemas y así aprender de sus compañeros de equipo.

El proyecto lleva poco tiempo de implementación por lo que no se pueden ver resultados económicos inmediatos, pero al largo plazo y según lo estimado sí lo habrá. Para el futuro se planea traer sellers extranjeros al Marketplace por lo que se seguirá utilizando este correo, se

le harán cambios mínimos como por ejemplo el idioma, y el umbral de tasas (ya que el seller extranjero será evaluado de manera distinta), pero el cuerpo y forma de enviar el correo será de la misma manera. También se seguirán automatizando procesos, pero siempre utilizando la API ya configurada para poder ver las métricas. Por ende, podemos decir que se creó una nueva forma de enviar correos en Walmart, que en el futuro seguirá utilizándose para diferentes campañas y gracias a esto es que surgieron muchas más ideas de cómo es que se podría seguir utilizando esta forma de enviar correos para el beneficio de la empresa, como por ejemplo utilizar este mismo método de envío de correos para enviar correos y reactivar a sellers que no venden hace 4 semanas para ofrecerles asesorías o para ser escuchados del porqué no venden o para invitarlos a volver a la plataforma y más proyectos que se siguen ideando día a día.

A nivel personal del alumno, esta experiencia le permitió desarrollar habilidades fundamentales para la carrera, ya que permitió desarrollar nuevas habilidades en el ámbito de la automatización de procesos incorporando datos desde otras plataformas, también permitió desarrollar habilidades en la medición de métricas, análisis y obtención de datos de una empresa de Marketing a nivel nacional e internacional donde existen millones de datos distintos.

La pasantía provocó al alumno reforzar el interés en seguir aprendiendo más para la carrera, seguir perfeccionándose en lo que le gusta y formar su propio camino en el campo de la informática. Esto generó entusiasmo para saber qué es lo que se podrá lograr en el futuro, para seguir contribuyendo en el desarrollo de nuevos proyectos y en el mundo de los datos a medida que va adquiriendo conocimiento con nuevas experiencias y desafíos laborales.

9 **Referencias:**

- Begnini Domínguez, L. F., Lecaro Lavayen, A. C., & Shauri Romero, J. D. (2022, 15 de julio). *Ventajas de la automatización de la gestión por procesos*. Recuperado de [<https://dspace.itsjapon.edu.ec/jspui/handle/123456789/3360>]
- García Ramírez, E. G. (2020). Impactos de la automatización en procesos laborales, percibidos por un grupo de líderes de gestión humana de una organización del sector manufacturero de la ciudad de Medellín. Recuperado de [https://repository.eafit.edu.co/bitstream/handle/10784/16065/EdwinGiovany_GarciaRamirez_2020.pdf?sequence=2&isAllowed=

- (Cámara de Comercio de Medellín, 2019). Impactos de la automatización en procesos laborales, percibidos por un grupo de líderes de gestión humana de una organización del sector manufacturero de la ciudad de Medellín. Recuperado de https://repository.eafit.edu.co/bitstream/handle/10784/16065/EdwinGiovany_GarciaRamirez_2020.pdf?sequence=2&isAllowed=y

10 Anexos:

Anexo 1: Columnas de las tablas con información en Google BigQuery:

Foto de columnas de la table OMS_STERLING_PURCHASE_ORDER:

The image shows two side-by-side screenshots of the Google BigQuery schema editor interface. Both screens have a header bar with 'Google Cloud' and 'wmt-edw-sandbox'. The left screen shows the first 20 columns of the table, while the right screen shows the remaining 25 columns. Each screenshot displays a table with columns for 'SCHEMA', 'DETAILS', 'PREVIEW', 'LINEAGE', and 'DATA PROFILE'. The 'SCHEMA' column lists the column names, which are underlined, indicating they are links to detailed views. The 'DETAILS' column shows the data type and mode (NULLABLE or NOT NULLABLE) for each column.

SCHEMA	DETAILS	PREVIEW	LINEAGE	DATA PROFILE
Field name	Type Mode			
PURCHASE_ORDER_NO	STRING NULLABLE			
SALES_ORDER_NO	STRING NULLABLE			
ORDER_DATE	DATETIME NULLABLE			
ORDER_TYPE	STRING NULLABLE			
PARTNER_ID	STRING NULLABLE			
SELLER_NAME	STRING NULLABLE			
MIN_DELIVERY_DATE	DATETIME NULLABLE			
MAX_DELIVERY_DATE	DATETIME NULLABLE			
EXPECTED_SHIPMENT_SHIPPED_DATE	DATETIME NULLABLE			
REQ_DELIVERY_DATE	DATETIME NULLABLE			
REQ_CANCEL_DATE	DATETIME NULLABLE			
REQ_SHIP_DATE	DATETIME NULLABLE			
DELIVERY_CHANNEL	STRING NULLABLE			
SELLER_EMAIL	STRING NULLABLE			
LABEL_URL	STRING NULLABLE			
STATUS	STRING NULLABLE			
STATUS_DESCRIPTION	STRING NULLABLE			
STATUS_REASON_CODE	STRING NULLABLE			
STATUS_DATE	DATETIME NULLABLE			
STATUS_QUANTITY	INTEGER NULLABLE			
LAST_SHIPMENT_STATUS	STRING NULLABLE			
ITEM_ID	STRING NULLABLE			
ITEM_DESCRIPTION	STRING NULLABLE			
PRIME_LINE_NO	INTEGER NULLABLE			
SUB_LINE_NO	INTEGER NULLABLE			
EXTN_SHIPPING_GROUP_NO	STRING NULLABLE			
TRACKING_NO	STRING NULLABLE			
CITY	STRING NULLABLE			
STATE	STRING NULLABLE			
SHIP_TO_KEY	STRING NULLABLE			
BILL_TO_KEY	STRING NULLABLE			
PURCHASE_ORDER_HEADER_KEY	STRING NULLABLE			
PURCHASE_ORDER_LINE_KEY	STRING NULLABLE			
SALES_ORDER_HEADER_KEY	STRING NULLABLE			
SALES_ORDER_LINE_KEY	STRING NULLABLE			
CREATSETS	DATETIME NULLABLE			
updateTable	STRING NULLABLE			

Qué se puede obtener de esta tabla:

- [OMS_STERLING_PURCHASE_ORDER](#): De esta tabla podemos obtener información de los estados de la orden al momento de comprar, se obtiene información tanto del courier de despacho (BlueExpress³¹), como también de la empresa de seguimiento de la orden (envíame³²). Obtenemos los datos de las ventas que fueron canceladas por: “SELLER REJECTION” (cancelación del seller), Error de precio, Agotado, Por operador, Por Operador, Por el operador antes de envío.”

Foto de columnas de la table OMS_STERLING_SALES_ORDER:

The image shows two side-by-side screenshots of the Google Cloud BigQuery interface, both titled "OMS_STERLING_SALES_ORDER". The left screenshot shows a simplified schema with 20 columns, while the right screenshot shows a detailed schema with 31 columns. Both schemas include columns like SALES_ORDER_NO, PURCHASE_ORDER_NO, RETURN_ORDER_NO, ORDER_DATE, ORDER_TYPE, PARTNER_ID, SELLER_NAME, ORDER_TOTAL_AMOUNT, ORDER_ORIGINAL_TOTAL_AMOUNT, MIN_DELIVERY_DATE, MAX_DELIVERY_DATE, EXPECTED_SHIPMENT_SHIPPED_DATE, REQ_DELIVERY_DATE, REQ_CANCEL_DATE, REQ_SHIP_DATE, STATUS, STATUS_DESCRIPTION, STATUS_REASON_CODE, STATUS_DATE, STATUS_QUANTITY, ITEM_ID, and various tax and payment-related fields.

Schema (Left)	Schema (Right)
SALES_ORDER_NO	ITEM_DESCRIPTION
PURCHASE_ORDER_NO	PRIME_LINE_NO
RETURN_ORDER_NO	SUB_LINE_NO
ORDER_DATE	SALES_CHANNEL
ORDER_TYPE	UNIT_PRICE
PARTNER_ID	ORDERED_QTY
SELLER_NAME	ORIGINAL_ORDERED_QTY
ORDER_TOTAL_AMOUNT	LINE_TOTAL
ORDER_ORIGINAL_TOTAL_AMOUNT	TAX
MIN_DELIVERY_DATE	TOTAL SHIPPING CHARGES
MAX_DELIVERY_DATE	TOTAL_DISCOUNT_CHARGES
EXPECTED_SHIPMENT_SHIPPED_DATE	OTHER_CHARGES_BALANCE
REQ_DELIVERY_DATE	PAYMENT_TYPE
REQ_CANCEL_DATE	EXTN_MP_COMMISSION
REQ_SHIP_DATE	EXTN_SHIPPING_GROUP_NO
STATUS	CITY
STATUS_DESCRIPTION	STATE
STATUS_REASON_CODE	RECAUDADOR_COLLECT_INVOICE_DOC_NO
STATUS_DATE	RECAUDADOR_COLLECT_INVOICE_DATE
STATUS_QUANTITY	RECAUDADOR_COLLECT_MONEYBACK_REQ_SENT
ITEM_ID	RECAUDADOR_RETURN_INVOICE_DOC_NO
	RECAUDADOR_RETURN_INVOICE_DATE

³¹ <https://www.blue.cl/>

³² <https://enviame.io/>

The screenshot shows the Google Cloud BigQuery schema browser interface. The top navigation bar includes 'Google Cloud' and 'wmt-edw-sandbox'. The main window displays the schema for the table 'OMS_STERLING_SALES_ORDER'. The schema consists of 20 columns:

	COLUMN	DATA TYPE	NULLABLE
1	EXTN_MP_COMMISSION	FLOAT	NULLABLE
2	EXTN_SHIPPING_GROUP_NO	STRING	NULLABLE
3	CITY	STRING	NULLABLE
4	STATE	STRING	NULLABLE
5	RECAUDADOR_COLLECT_INVOICE_DOC_NO	STRING	NULLABLE
6	RECAUDADOR_COLLECT_INVOICE_DATE	STRING	NULLABLE
7	RECAUDADOR_COLLECT_MONEYBACK_REQ_SENT	STRING	NULLABLE
8	RECAUDADOR_RETURN_INVOICE_DOC_NO	STRING	NULLABLE
9	RECAUDADOR_RETURN_INVOICE_DATE	STRING	NULLABLE
10	RECAUDADOR_RETURN_MONEYBACK_REQ_SENT	STRING	NULLABLE
11	WALMART_ID	STRING	NULLABLE
12	SHIP_TO_KEY	STRING	NULLABLE
13	BILL_TO_KEY	STRING	NULLABLE
14	SALES_ORDER_HEADER_KEY	STRING	NULLABLE
15	SALES_ORDER_LINE_KEY	STRING	NULLABLE
16	PURCHASE_ORDER_LINE_KEY	STRING	NULLABLE
17	PURCHASE_ORDER_HEADER_KEY	STRING	NULLABLE
18	RETURN_ORDER_HEADER_KEY	STRING	NULLABLE
19	RETURN_ORDER_LINE_KEY	STRING	NULLABLE
20	CREATESETS	DATETIME	NULLABLE
21	updateTable	STRING	NULLABLE

At the bottom of the schema browser, there are buttons for 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES'.

Qué se puede obtener de esta tabla:

- OMS_STERLING_SALES_ORDER: De esta tabla obtenemos información de la orden en términos de venta, comisiones e información financiera.

Foto de columnas de la tabla OMS_STERLING_RETURN_ORDER:

Field name	Type	Mode
RETURN_ORDER_NO	STRING	NULLABLE
SALES_ORDER_NO	STRING	NULLABLE
ORDER_DATE	DATETIME	NULLABLE
ORDER_TYPE	STRING	NULLABLE
PARTNER_ID	STRING	NULLABLE
SELLER_NAME	STRING	NULLABLE
STATUS	STRING	NULLABLE
STATUS_DESCRIPTION	STRING	NULLABLE
LAST_RETURN_STATUS	STRING	NULLABLE
LAST_RETURN_STATUS_DESCRIPTION	STRING	NULLABLE
STATUS_DATE	DATETIME	NULLABLE
STATUS_QUANTITY	INTEGER	NULLABLE
ITEM_ID	STRING	NULLABLE
ITEM_DESCRIPTION	STRING	NULLABLE
PRIME_LINE_NO	INTEGER	NULLABLE
SUB_LINE_NO	INTEGER	NULLABLE
RETURN_TRACKING_NO	STRING	NULLABLE
TRACKING_NO	STRING	NULLABLE
EXTN_RETURN_MODE	STRING	NULLABLE
RETURN_REASON_CODE	STRING	NULLABLE
RETURN_REASON_DESCRIPTION	STRING	NULLABLE

Field name	Type	Mode
STATUS_DATE	DATETIME	NULLABLE
STATUS_QUANTITY	INTEGER	NULLABLE
ITEM_ID	STRING	NULLABLE
ITEM_DESCRIPTION	STRING	NULLABLE
PRIME_LINE_NO	INTEGER	NULLABLE
SUB_LINE_NO	INTEGER	NULLABLE
RETURN_TRACKING_NO	STRING	NULLABLE
TRACKING_NO	STRING	NULLABLE
EXTN_RETURN_MODE	STRING	NULLABLE
RETURN_REASON_CODE	STRING	NULLABLE
RETURN_REASON_DESCRIPTION	STRING	NULLABLE
CITY	STRING	NULLABLE
STATE	STRING	NULLABLE
SHIP_TO_KEY	STRING	NULLABLE
BILL_TO_KEY	STRING	NULLABLE
RETURN_ORDER_HEADER_KEY	STRING	NULLABLE
RETURN_ORDER_LINE_KEY	STRING	NULLABLE
SALES_ORDER_HEADER_KEY	STRING	NULLABLE
SALES_ORDER_LINE_KEY	STRING	NULLABLE
CREATETS	DATETIME	NULLABLE
updateTable	STRING	NULLABLE

Qué se puede obtener de esta tabla:

- OMS_STERLING_RETURN_ORDER: De esta tabla podemos obtener información de los estados de devolución de la orden. En esta tabla encontramos el detalle del por qué se devolvió y el status de la devolución. Y también obtenemos el valor de “Devolución concluida” el cual nos indica que el proceso de devolución fue completado.

Anexo 2: Objetivos SMART:

Por otra parte, definiremos objetivos del proyecto mediante la metodología SMART, a continuación, se muestra el detalle de cada uno:

S: El objetivo general específico de este proyecto es crear un modelo que permita reducir los tiempos de obtención, visualización y análisis de los datos para posteriormente enviarlos a cada seller para su correcta visualización, sin que existan fugas de información y así poder evitar que avancen de strike, y también que la misma empresa pueda obtener datos del comportamiento de los sellers de manera fácil, ordenada y precisa.

M: El objetivo medible será el poder reducir la cantidad de seller que avanzan semanalmente de strike, reducir los tiempos de envío de correos y obtención de datos, si utilizamos toda la información que Walmart tiene de los sellers y sus ventas se puede lograr crear un feedback y un análisis para estos, esto dará por logrado cuando cada seller logre recibir sus KPI de manera que este pueda visualizarlos y leer un análisis de su comportamiento para que comprendan que es lo que pasará en un futuro si es que sus ventas siguen siendo positivas/negativas (va a depender del caso).

A: El objetivo alcanzable de este proyecto es que se logre crear un modelo, para mejorar el servicio al seller, ya que las herramientas con las que se está trabajando actualmente son herramientas con las cuales se ha trabajado con anterioridad y sumándole las herramientas propias que tiene Walmart se pueden crear cosas dentro del mismo entorno laboral.

R: El objetivo relevante de este proyecto es evitar la pérdida de sellers, debido a que mientras los sellers más avancen de strike es posible que alguno pueda ser dado de baja y así la empresa pierde oportunidades de venta y por ende pierde comisiones, si se entrega información a los sellers de manera completa, periódica y segura, esto logrará que los sellers puedan tener una guía semanal/mensual de sus ventas para evitar avanzar de strike.

T: El objetivo de tiempo del proyecto es que se espera que el proyecto se complete a más tardar en el mes de octubre, ya que de esta manera se podrán obtener feedback y datos para posteriormente comenzar a obtener resultados y hacer un análisis/medición de los datos obtenidos, agregándole que también se hizo una carta Gantt en la que se definió la fecha límite del proyecto.

Anexo 3: Tabla creada en Google Big Query para tasa de cancelación

Esta es la tabla que se creó en Google Big Query y posteriormente se filtra en el código de Python por cada seller para enviar los datos correspondientes a cada uno.

Que nos muestra la tabla de cancelaciones:

Google Cloud Search: SS_CASO1.5 FINAL FINAL

Query results

Row	ISO_WEEK	ISO_YEAR	PARTNER_ID	SELLER_NAME	LINEAS_TOTALES	LINEAS_CANC_OP	LINEAS_CANC_SELL	TASA_CANCELACION
1	44	2022			8	0	0	0
2	44	2022			7	0	0	0
3	44	2022			5	0	0	0
4	44	2022			4	0	0	0
5	44	2022			4	0	0	0
6	44	2022			2	0	0	0
7	44	2022			2	0	0	0
8	44	2022			2	0	0	0
9	44	2022			2	0	0	0
10	44	2022			2	0	0	0
11	44	2022			1	0	0	0
12	44	2022			1	0	0	0

Results per page: 50 | 1 - 50 of 12067 | Refresh

Como es el código para la tabla de cancelaciones:

```

1 WITH
2   SALES_ORDER AS(
3     SELECT DISTINCT
4       SALES_ORDER_NO,
5       ORDER_DATE AS ORDER_DATETIME,
6       FORMAT_DATETIME("%Y-%m-%d", ORDER_DATE) ORDER_DATE,
7       FORMAT_DATETIME("%H:%M", ORDER_DATE) ORDER_HOUR,
8       PARTNER_ID,
9       SELLER_NAME,
10      PRIME_LINE_NO,
11      SUB_LINE_NO,
12      TRIM(ITEM_ID) AS ITEM_ID,
13      ITEM_DESCRIPTION,
14      ORDERED_QTY,
15      SALES_ORDER_LINE_KEY,
16      PURCHASE_ORDER_LINE_KEY,
17      RETURN_ORDER_LINE_KEY
18  FROM
19    (SELECT *, RANK() OVER (PARTITION BY SALES_ORDER_NO, PRIME_LINE_NO, SUB_LINE_NO ORDER BY STATUS_DESC, STATUS_DATE DESC) AS RNK
20  FROM `wmt-edw-sandbox.WM_AD_HOC_K2_0000.OMS_STERLING_SALES_ORDER`
21  )
22 WHERE
23   RNK = 1
24   AND
25   PARTNER_ID NOT IN (
26     SELECT
27       CAST(PARTNER_ID AS STRING)
28     FROM
29       `WM_AD_HOC_K2_0000.LISTADO_SELLERS_PRUEBA`
30   ),
31   ),
32 PURCHASE_ORDER AS(
33   SELECT DISTINCT
34     PURCHASE_ORDER_NO,
35     TRACKING_NO,
36     STATUS,
37     STATUS_DESCRIPTION,
38     STATUS_REASON_CODE,
39     FORMAT_DATETIME("%Y-%m-%d", STATUS_DATE) STATUS_DATE,
40     FORMAT_DATETIME("%H:%M", STATUS_DATE) STATUS_HOUR,
41     LAST_SHIPMENT_STATUS,
42     LAST_SHIPMENT_STATUS_DESCRIPTION,
43     LAST_SHIPMENT_STATUS_DATE,
44     SALES_ORDER_LINE_KEY
45 )

```

Ln 5, Col 42 Spaces: 4 bq.cancelaciones.sql

```

45 SALES_ORDER_LINE_KEY
46
47   (SELECT *, RANK() OVER (PARTITION BY PURCHASE_ORDER_NO, PRIME_LINE_NO, SUB_LINE_NO ORDER BY STATUS DESC, STATUS_DATE DESC) AS RNK
48   FROM `wmt-edw-sandbox.WP_AD_HOC_K2_0000.OMS_STERLING_PURCHASE_ORDER`
49 )
50 WHERE
51   RNK = 1
52 ),
53
54 RETURN_ORDER AS (
55   SELECT DISTINCT
56     RETURN_ORDER_NO,
57     FORMAT_DATETIME("%Y-%m-%d", ORDER_DATE) ORDER_DATE,
58     FORMAT_DATETIME("%H:%M", ORDER_DATE) ORDER_HOUR,
59     PRIME_LINE_NO,
60     SUB_LINE_NO,
61     STATUS,
62     STATUS_DESCRIPTION,
63     FORMAT_DATETIME("%Y-%m-%d", STATUS_DATE) STATUS_DATE,
64     FORMAT_DATETIME("%H:%M", STATUS_DATE) STATUS_HOUR,
65     LAST_RETURN_STATUS,
66     LAST_RETURN_STATUS_DESCRIPTION,
67     EXTN_RETURN_MODE,
68     RETURN_REASON_CODE,
69     RETURN_TRACKING_NO,
70     SALES_ORDER_LINE_KEY
71
72   FROM (
73     SELECT *, RANK() OVER (PARTITION BY RETURN_ORDER_NO, PRIME_LINE_NO, SUB_LINE_NO ORDER BY STATUS DESC, STATUS_DATE DESC) AS RNK
74     FROM `wmt-edw-sandbox.WP_AD_HOC_K2_0000.OMS_STERLING_RETURN_ORDER`
75   )
76 WHERE
77   RNK = 1
78 )
79
80 SELECT
81   CAST(EXTRACT(ISO_WEEK FROM SO.ORDER_DATETIME) AS STRING FORMAT "00") AS ISO_WEEK,
82   EXTRACT(ISOYEAR FROM SO.ORDER_DATETIME) AS ISO_YEAR,
83   SO.PARTNER_ID AS PARTNER_ID,
84   SO.SELLER_NAME AS SELLER_NAME,
85   COUNT(SO.PARTNER_ID) AS LINEAS_TOTALES,
86   COUNT(
87     CASE WHEN LOWER(PO.STATUS_REASON_CODE) IN ('por operador', 'por el operador antes del envío') THEN SO.PARTNER_ID
88     ELSE NULL END
89   ) AS LINEAS_CAN_OP,

```



```

89 ) AS LINEAS_CAN_OP,
90
91 COUNT(
92   CASE WHEN LOWER(PO.STATUS_REASON_CODE) IN ('seller rejection', 'error de precio', 'agotado') THEN SO.PARTNER_ID
93   ELSE NULL END
94 ) AS LINEAS_CAN_SELLER,
95
96 --CONCAT(
97   CAST(
98     ROUND(
99       SAFE_DIVIDE(COUNTIF(PO.STATUS = "9000", COUNT(SO.PARTNER_ID))
100      , 4) -- REDONDEO PARA LA DIVISION
101      * 100) -- PARA EL CALCULO DEL %
102      AS STRING FORMAT "999.0" -- CAST
103      , "%") -- CONCAT
104      AS TASA_CANCELACION
105
106 FROM SALES_ORDER SO
107 LEFT JOIN PURCHASE_ORDER PO ON SO.SALES_ORDER_LINE_KEY = PO.SALES_ORDER_LINE_KEY
108 LEFT JOIN RETURN_ORDER RO ON SO.SALES_ORDER_LINE_KEY = RO.SALES_ORDER_LINE_KEY
109
110 WHERE
111   CAST(SO.ORDER_DATE AS DATE) >= DATE('2022-11-01')
112   AND
113   SO.PARTNER_ID NOT IN ("10001238998")
114 GROUP BY
115   ISO_WEEK, ISO_YEAR, PARTNER_ID, SELLER_NAME
116 ORDER BY
117   ISO_YEAR ASC, ISO_WEEK ASC, LINEAS_TOTALES DESC

```

Anexo 4: Tabla creada en Google Big Query para tasa de cancelación

Esta es la tabla que se creó en Google Big Query y posteriormente se filtra en el código de Python por cada seller para enviar los datos correspondientes a cada uno.

Tabla de devoluciones:

Google Cloud Search / for resources, docs, products, and more Search

SS_CASO2 RUN SAVE QUERY SHARE SCHEDULE MORE Query completed.

Press Alt+F1 for Accessibility Options.

Query results

Row	YEAR	MONTH	PARTNER_ID	SELLER_NAME	LINEAS_TOTALES	ORDENES_DEVUELTA	TASA_DEVOLUCION
1	2023	11			104	4	3.85 %
2	2023	11			312	3	0.96 %
3	2023	11			1482	38	2.56 %
4	2023	11			384	8	2.08 %
5	2023	11			13	0	0 %
6	2023	11			1628	35	2.15 %
7	2023	11			192	1	0.52 %
8	2023	11			106	1	0.94 %
9	2023	11			44	0	0 %
10	2023	11			22	0	0 %
11	2023	11			63	0	0 %
12	2023	11			115	1	0.87 %

Results per page: 50 ▾ 1 – 50 of 1064 | REFRESH

Como es el código para la tabla de devoluciones:

```

1 DECLARE DATE_TODAY DATE DEFAULT CURRENT_DATE();
2 DECLARE MONTH_DEFAULT(EXTRACT(MONTH FROM DATE_TODAY ));
3 DECLARE THIS_YEAR DEFAULT(EXTRACT(ISOLEVEL FROM DATE_TODAY ));
4
5 WITH P AS (
6   SELECT
7     ORDER_DATE,
8     EXTRACT(MONTH FROM ORDER_DATE) AS MONTH,
9     EXTRACT(ISOLEVEL FROM ORDER_DATE) AS YEAR,
10    PARTNER_ID,
11    SELLER_NAME,
12    SALES_ORDER_LINE_KEY
13   FROM (SELECT *, RANK() OVER
14     (PARTITION BY PURCHASE_ORDER_NO, PRIME_LINE_NO, SUB_LINE_NO ORDER BY STATUS DESC, STATUS_DATE DESC) AS RNK
15   FROM `WM_AD_HOC_K2_0000.OMS_STERLING_PURCHASE_ORDER` AS P
16 )
17 WHERE
18   RNK = 1
19 ),
20
21 R AS (
22   SELECT
23     PARTNER_ID,
24     SELLER_NAME,
25     STATUS_DESCRIPTION,
26     SALES_ORDER_LINE_KEY
27   FROM (SELECT *, RANK() OVER
28     (PARTITION BY RETURN_ORDER_NO, PRIME_LINE_NO, SUB_LINE_NO ORDER BY STATUS DESC, STATUS_DATE DESC) AS RNK
29   FROM `WM_AD_HOC_K2_0000.OMS_STERLING_RETURN_ORDER` )
30 )
31
32 SELECT
33   P.YEAR,
34   P.MONTH,
35   P.PARTNER_ID,
36   P.SELLER_NAME,
37   COUNT(P.PARTNER_ID) AS LINEAS_TOTALES,
38   COUNTIF(R.STATUS_DESCRIPTION IN ('Devolución Concluida')) AS ORDENES_DEVUeltas,
39
40   CAST(
41     ROUND(
42       SAFE_DIVIDE(
43         (COUNTIF(R.STATUS_DESCRIPTION IN ('Devolución Concluida')) * 100 ), COUNT(P.PARTNER_ID)), 4) AS STRING) AS TASA_DEVOLUCION
44
45 FROM P
46 LEFT JOIN R ON P.SALES_ORDER_LINE_KEY = R.SALES_ORDER_LINE_KEY

```

Saving completed

Ln 43, Col 100 Spaces: 4 bq_devoluciones.sql

Tabs Settings Help | Mem: 390 / 10000 MB CPU: 0.41% / 500%

bq_devoluciones.sql

```
42     SAFE_DIVIDE(
43         (COUNTIF(R.STATUS_DESCRIPTION IN ('Devolución Concluida')) * 100 ), COUNT(P.PARTNER_ID)), 4) AS STRING) AS TASA_DEVOLUCION
44     FROM P
45     LEFT JOIN R ON P.SALES_ORDER_LINE_KEY = R.SALES_ORDER_LINE_KEY
46     WHERE
47         --P.PARTNER_ID = '10001263343'
48         --AND
49         CAST(P.ORDER_DATE AS DATE) > DATE_SUB(CURRENT_DATE(), INTERVAL 1 MONTH)
50     AND
51     P.PARTNER_ID NOT IN (
52         SELECT
53             partner_id
54             FROM
55                 wmt-edw-sandbox.WM_AD_HOC_K2_000D.LISTADO_SELLERS_PRUEBA
56         )
57     GROUP BY
58     P.YEAR, P.MONTH, P.PARTNER_ID, P.SELLER_NAME
59     ORDER BY
60     P.MONTH DESC, P.YEAR DESC
61     --LIMIT 1
62
```

Saving completed

Ln 43, Col 100 Spaces: 4 bq_devoluciones.sql

Anexo 5 Metodologías:

En este punto se describirán las metodologías a utilizar para la implementación del proyecto, las metodologías que se utilizarán serán principalmente la metodología AGILE y SCRUM , pero también las que las que se han aprendido en la universidad y en la empresa.

Metodología AGILE:



Imagen 3: Método AGILE, **Fuente:** <https://www.northware.mx/blog/desarrollo-en-cascada-waterfall-vs-desarrollo-agile-scrum/>

Metodología SCRUM:

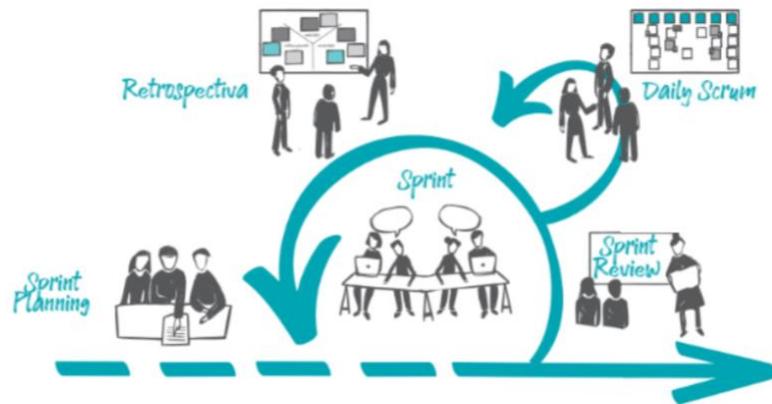


Imagen 4: Método SCRUM, **Fuente:** <https://caroli.org/es/scrum-significado-aplicacion-conceptos-y-ejemplos/>

Como se puede ver en las imágenes, estas metodologías consisten en desarrollar “N” funcionalidades en ciclos o sprints para este caso, a diario se tienen reuniones para poder

comentar que es lo que se está haciendo y pedir orientación al equipo en caso de ser necesario, luego al final de cada sprint se muestra el proyecto que se desarrolló y el equipo entrega un feedback con mejoras o cambios que se podrían implementar antes de pasar a la etapa de test y de implementación. Esta es una metodología bastante flexible, ya que como se dijo, hay que estar abierto a implementar mejoras en beneficio del cliente o la empresa, también se está constantemente en contacto con el equipo, por lo que hay mucha colaboración en caso de ser necesario y por ende, existe una retroalimentación muy continua.

También es importante destacar que todas las metodologías se trabajaran dentro de los estándares éticos y de la calidad que establece la empresa Walmart Chile y la Universidad Adolfo Ibáñez, garantizando así una correcta privacidad de los datos y el cumplimiento de las distintas regulaciones que se piden.

Anexo 6: Matriz de riesgos:

Riesgo	Descripción	Probabilidad	Impacto	Prioridad	Mitigación
Problemas de compatibilidad	Al ser Walmart una empresa tan estricta con el tema de la seguridad de datos es muy importante utilizar servicios compatibles con la empresa para evitar problemas de este tipo.	Medio	Alto	Moderado	Realizar pruebas de compatibilidad antes de la implementación del proyecto.
Fallos en la automatización o caídas del servicio	Errores en el proceso de automatización o caídas de los servicios utilizados para el envío de los correos.	Bajo	Medio	Moderado	Implementar controles de calidad, "planes B" y hacer pruebas durante el proceso.
Seguridad de datos	Alguna mala práctica dentro del código podría filtrar o enviar correos erróneos a los sellers. Y también malas prácticas dentro de Walmart, ya que en la empresa existe un departamento dedicado a la protección y seguridad de los datos contra posibles amenazas.	Bajo	Muy Alto	Medio	Implementar medidas de seguridad robustas y cifrado de datos. No compartir directamente datos confidenciales de la empresa a sitios web externos.
Falta de recursos	Walmart tiene muchas licencias ya disponibles para el trabajador, por lo que proponer comprar un servicio externo de pago es muy complicado.	Bajo	Medio	Moderado	Crear un plan de proyecto realista con los servicios que nos ofrece Walmart y evitar el uso de servicios externos.
Cambios en las leyes de manejo de datos	Cambios en las leyes de privacidad de datos o regulaciones que puedan afectar directamente el proyecto.	Muy Bajo	Muy Alto	Medio	Estar al tanto de lo que pasa y en caso de haber algún cambio implementar los respectivos cambios.
Atraso por parte mía	Limitaciones de tiempo, falta de conocimiento que implique dedicar tiempo a estudiar cosas nuevas y/o caídas del servicio de internet.	Medio	Medio	Medio	Asegurar una asignación adecuada de tiempos para evitar hacer cosas apuradas y prevenir posibles caídas de internet.

Tabla 2: Matriz de riesgos y estrategias de mitigación del proyecto. Fuente: Elaboración propia

		Impacto					
		Muy bajo	Bajo	Medio		Alto	Muy alto
Probabilidad	Muy baja	Muy bajo	Muy bajo	Bajo		Medio	Medio
	Baja	Muy bajo	Bajo	Moderado		Medio	Medio
	Medio	Bajo	Moderado	Medio		Moderado	Alto
	Alta	Medio	Medio	Moderado		Alto	Muy alto
	Muy alta	Medio	Medio	Alto		Muy alto	Muy alto

Tabla 3: Matriz de riesgos Fuente: Elaboración propia

Anexo 7: Plan de implementación

Es muy importante mencionar que todos estos objetivos están siendo cumplidos bajo la planificación de una carta Gantt creada al inicio del semestre.

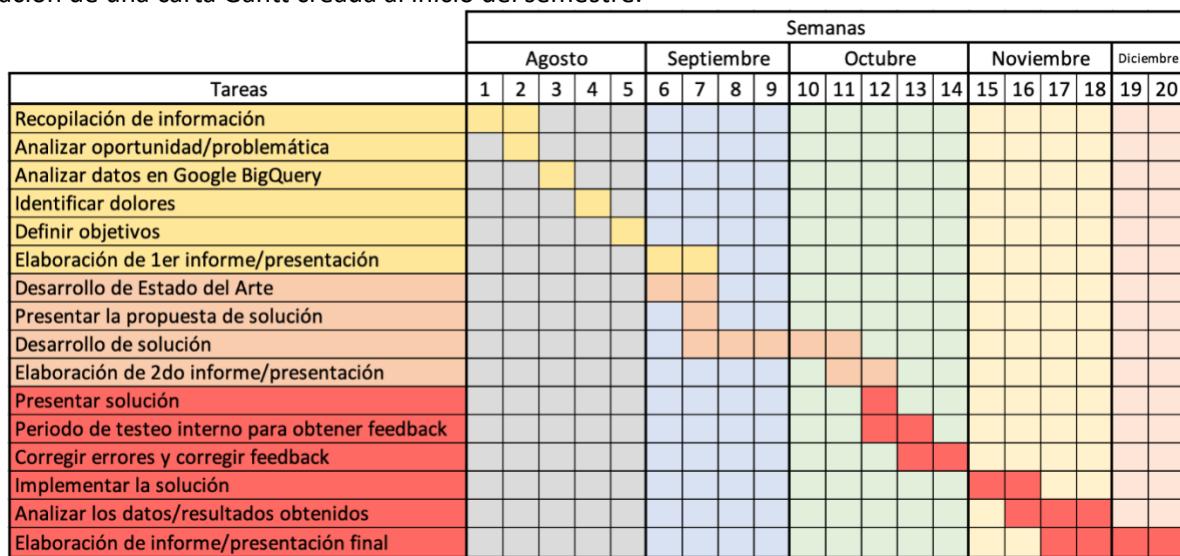


Imagen 21: Carta Gantt Fuente: Elaboración propia

Los colores de las tareas en la carta Gantt representan:

1. Amarillo: Periodo de estudio, entendimiento y aprendizaje dentro de la empresa
2. Naranjo: Periodo de desarrollo técnico de la solución

3. Rojo: Periodo final del proyecto, donde se presentará la solución, se probará y se implementará.

El proyecto está planeado para implementarse de la siguiente manera:

En primer lugar se hará un análisis y estudio de cómo funciona el proceso de envío de los correos en la actualidad, entendiendo el proceso, analizando la documentación y escribiendo las posibles mejoras dentro del servicio para posteriormente analizarlas y crear un modelo que sea adecuado para lo que la empresa necesita.

- **Agosto:** Mes de “onboarding”³³. Fue en este mes que se me presentó la oportunidad que existía y se me recomendó que revisara documentación de cómo funciona actualmente el proceso. A final de este mes se me comenzaron a asignar pequeños trabajos para que fuera entendiendo como es que funcionaba el proceso de una perspectiva en primera persona (el envío de los correos, la actualización de tabla de strikes y la manipulación de las bases de datos de la empresa).
- **Septiembre:** En este mes se comenzó a trabajar completamente en el proyecto, aplicando todo lo realizado y aprendido durante el onboarding del mes anterior, se crearon propuestas de solución, y luego de hacer un análisis de cada una se decidió por la propuesta seleccionada y se presentó a la empresa. Posteriormente en este mes se comenzó con el desarrollo del proyecto.
- **Octubre:** Durante este mes se sigue con el desarrollo y se planea presentar la solución creada a la empresa, comenzar un periodo de prueba interno para implementar mejoras necesarias y/o obtener feedback de parte del equipo de trabajo, corregir errores y posteriormente implementar la solución al final del mes.
- **Noviembre:** Ya para comenzar a medir el impacto en este mes se implementó la solución para poder obtener resultados y hacer un análisis robusto del impacto del proyecto.

³³ Proceso de incorporación y capacitación dentro de la empresa, en Walmart la duración de este proceso es de 1 mes aproximadamente.

- **Diciembre:** Por último en este mes se comienza a cerrar el periodo de análisis y se comenzará a elaborar el informe final para comenzar a crear las conclusiones y discusión.

Anexo 8: Contrato de confidencialidad

Ha firmado: "Convenio de practica_anexo TT_anexo de confidencialidad_M" ➤ Recibidos x

 Walmart Stores <echosign@echosign.com>
para Valentina, mi ✓ jue, 3 ago, 11:17



Ha acabado de firmar
Convenio de practica_anexo TT_anexo de confidencialidad_M

[Abrir el acuerdo](#)

Se ha adjuntado el acuerdo final para su consulta. Leer con **Acrobat Reader**. También puede [abrirlo en línea](#) para revisar el historial de actividad.

RE: Proyecto de título - Felipe Cornejo 📲

 Etica Chile
Para: Felipe Cornejo - Vendor; Sebastian Castillo (Chile)
CC: María Muñoz (Chile)

Mar 14/11/2023 20:54

Estimado Felipe,
Gracias por contactarte con la Oficina de Ética.
Te comentamos que nuestro Código de Conducta señala que debemos siempre proteger la información comercial y personal de acceso y divulgación de Walmart y en relación con esto, luego de analizar tu consulta, desde la Oficina de Ética no vemos inconveniente en que puedas realizar tu proyecto de título, siempre y cuando tengas presente que no puedes utilizar datos e información que pueda ser considerada como sensible, estratégica y confidencial. En el caso de utilizar alguna cifra como venta, indicador, etc., debes multiplicarla por un factor común y utilizar el dato como referencial, al igual que los datos que mencionas de los Sellers, que deben ser reemplazados.
Ponemos en copia a Sebastian Castillo, Abogado del Área Legal, para que pueda entregarte más recomendaciones y si es necesario la firma de un NDA.
Cualquier duda nos avísas.

Saludos,

Oficina de Ética
Teléfono: 800 55 07 07
E-mail: etica.chile@walmart.com

Walmart Chile
Conozca nuestra [Declaración de Ética Global](#)
En Walmart **no permitimos represalias** en contra de colaboradores que plantean inquietudes o preguntas sobre una mala conducta.

Anexo 9: Código tasa de cancelación

The screenshot shows two Jupyter Notebook sessions running on a Linux system (Ubuntu 18.04 LTS) with Python 3.6. The top session, titled 'salesforce_v4.ipynb', contains code for interacting with Salesforce using pandas and requests. It includes imports for pretty_html_table, pandas, numpy, requests, json, os, re, time, datetime, pytz, and google.cloud.bigquery. It defines a timezone for Santiago, Chile, calculates five weeks ago, and uses iso calendar to generate a year-week identifier. It also imports credentials from a JSON file and connects to BigQuery using a client from the google.cloud library. The bottom session, also titled 'salesforce_v4.ipynb', contains code for extracting seller contact emails from BigQuery. It imports google, os, and bqclient libraries, sets environment variables, and defines a function df_correos_biggquery() that reads a query file, executes it against BigQuery, and returns a DataFrame. It also defines a function df_con_contactos() that concatenates multiple contact types into a single column.

```
[36]: # Para mejorar la visualización de la tabla usamos la librería pretty_html_table
from pretty_html_table import build_table

# Pandas para el manejo de tablas y datos
import pandas as pd
import numpy as np

# Librerías que utilizaremos para hacer consultas hacia Salesforce
import requests as rq
import json

# Para utilizar archivos
import os

# Expresiones regulares para definir formato de casilla mail
import re

# Importamos time.sleep para el envío de mails
from time import sleep

# Definimos la timezone para obtener el dia/hora en caso de necesitarlo
from datetime import datetime, timedelta

import pytz

santiago_tz = pytz.timezone('America/Santiago') # En este caso la de Santiago, Chile
today = datetime.now(tz=santiago_tz)

five_weeks_ago = (today - timedelta(days=35)).isocalendar()[1]
year_five_weeks_ago = (today - timedelta(days=35)).isocalendar()[0]

iso_calendar = today.isocalendar()
iso_year = iso_calendar[0]
iso_week = iso_calendar[1]
day_of_week = iso_calendar[2]
year_week_act = str(iso_year) + str(iso_week)
year_week_leesfives = str(year_five_weeks_ago) + str(five_weeks_ago)

# Librería para hacer queries y demás a Big Query.
from google.cloud import bigquery

# Importamos las credenciales
key_path = "/home/jupyter/Seller-Performance-Felipe/credentials.json"

[37]: # Esta función hace una query a Big Query y devuelve un dataframe con datos de correos de sellers.

def df_correos_biggquery():

    # Leemos el contenido del archivo dado por la ruta.
    with open(query_correos_path, 'r') as file:
        query_correos = file.read()

    df_correos = client.query(query_correos).to_dataframe().astype(str)

    # Inyección de la query al cliente de Big Query y transformación a DataFrame.

    return df_correos

[38]: # Esta función nos permite obtener todos los mail de contacto de los sellers.

def df_con_contactos():
    df_correos = df_correos_biggquery()
    df_correos[CORREO_FINAL] = None
    # Armamos un DataFrame con los correos que se pueden encontrar en Salesforce.

    lst_cols = [
        'Correo_Representante_Legal',
        'Correo_Contacto_Comercial',
        'Correo_Contacto_Logistica',
        'Correo_Contacto_SAC',
        'Correo_Contacto_Otros',
        'Correo_Contacto_Capacitaciones',
        'Correo_Contacto_Performance',
        'Correo_Contacto_Otro'
    ]
```

The screenshot shows two Jupyter Notebook sessions side-by-side, both titled "salesforce_v4.ipynb".

Top Notebook (Cell 39):

```

# Esta función nos permite obtener los datos de las columnas que tendrá la tabla que se muestra en el correo.
def bbdd_resumen_ordenes():

    # Leemos el contenido del archivo
    with open(query_bq_cancelaciones_path, 'r') as file:
        query_ordenes = file.read()

    dict_valores = {
        'PARTNER_ID': str,
        'ISO_YEAR': np.int16,
        'PARTNER_ID': np.int64,
        'SELLER_NAME': str,
        'LINEAS_TOTALES': np.int16,
        'LINEAS_CANC_OP': np.int16,
        'LINEAS_CANC_SELLER': np.int16,
        'TASA_CANCELACION': np.float32
    }

    # Proceso de ejecución en BigQuery
    df_ordenes = client.query(query_ordenes).to_dataframe(dtypes = dict_valores)

    # Vaciamos los espacios que quedan a los costados (por defecto bigquery les asigna unos espacios)
    df_ordenes[["ISO_WEEK"]] = df_ordenes[["ISO_WEEK"]].apply(lambda x: x.strip())

    # Creamos columna year_week
    df_ordenes[["YEAR_WEEK"]] = df_ordenes[["ISO_YEAR"]].astype(str) + df_ordenes[["ISO_WEEK"]].astype(str)

    return df_ordenes

```

Bottom Notebook (Cell 40):

```

# Con lo anterior obtenemos la tabla completa, ya que filtraremos los DF fuera de BigQuery

# Obtenemos los datos de venta
df_resumen_ordenes = bbdd_resumen_ordenes()

# df con los correos de los sellers
df_correos = df_com_correos()

# Sacamos la semana actual
df_resumen_ordenes = df_resumen_ordenes[df_resumen_ordenes['YEAR_WEEK'] != year_week_act]

# Armamos un listado con sellers que han vendido durante al menos las ultimas 4 semanas
lst_sellers_con_venta = df_resumen_ordenes[df_resumen_ordenes['YEAR_WEEK'].astype(int) >= int(year_week_lessfive)][['PARTNER_ID']].unique().tolist()

# Ordenamos el df
df_resumen_ordenes.sort_values(by = ['PARTNER_ID', 'YEAR_WEEK'], ascending = [1, 0], inplace = True)

# Obtenemos los sellers que han vendido algún item en las últimas 4 semanas
def get_last_4_weeks(df_resumen_ordenes, lst_sellers_con_venta):
    df = pd.DataFrame(columns = df_resumen_ordenes.columns)

    for pid in lst_sellers_con_venta:
        df = pd.concat([df, df_resumen_ordenes[df_resumen_ordenes['PARTNER_ID'] == pid].head(4)])

```

Bottom Notebook (Cell 41):

```

    return df

# Acá se comienza a filtrar la tabla
# Toma las ultimas 4 semanas con venta (sin orden).
df_definitivo = get_last_4_weeks(df_resumen_ordenes, lst_sellers_con_venta)

# Agregamos el simbolo de % y corregimos decimales en 0% y 100%

```

File Edit View Run Kernel Git Tabs Settings Help | Mem: 228 / 10000 MB CPU: 0.2% / 500%

salesforce_v4.ipynb

```
# Agregamos el simbolo de % y corregimos decimales en 0% y 100%
df_definitivo['TASA_CANCELACION'] = df_definitivo['TASA_CANCELACION'].astype(float)
df_definitivo['TASA_CANCELACION'] = df_definitivo['TASA_CANCELACION'].map('{:.1f}%'.format)
df_definitivo['TASA_CANCELACION'] = df_definitivo['TASA_CANCELACION'].replace(to_replace = '0.0%', value = '0%')
df_definitivo['TASA_CANCELACION'] = df_definitivo['TASA_CANCELACION'].replace(to_replace = '100.0%', value = '100%')

dict_rename = {
    'ISO_WEEK': 'Semana',
    'ISO_YEAR': 'Año',
    'PARTNER_ID': 'PARTNER_ID',
    'SELLER_NAME': 'SELLER_NAME',
    'LINEAS_TOTALES': 'Lineas totales',
    'LINEAS_CANC_OP': 'Lineas canceladas por Walmart',
    'LINEAS_CANC_SELLER': 'Lineas canceladas por ti',
    'TASA_CANCELACION': 'Tasa de cancelación (%)'
}

# Renombramos las columnas para que sea más amigable para el seller
df_definitivo = df_definitivo.rename(dict_rename, axis = 1)

# Recortamos el df
df_definitivo = df_definitivo[list(dict_rename.values())]

# Convertimos el df a string para poder utilizarlo en el ciclo for
df_definitivo = df_definitivo.astype(str).sort_values(by = ['PARTNER_ID', 'Año', 'Semana'], ascending = [1, 0, 0]).reset_index(drop = True)

columnas_finales = [
    'Semana',
    'Año',
    'Lineas totales',
    'Lineas canceladas por Walmart',
    'Lineas canceladas por ti',
    'Tasa de cancelación (%)'
]
df_definitivo
```

[43]:

	Semana	Año	PARTNER_ID	SELLER_NAME	Lineas totales	Lineas canceladas por Walmart	Lineas canceladas por ti	Tasa de cancelación (%)
0	46	2023	.		25	0	0	0%
1	45	2023	.		23	0	0	0%
2	44	2023			12	0	0	0%
3	43	2023	.		14	0	0	0%

Simple 0 1 Default_env_61503 | Idle Saving completed Mode: Command | Mem: 228 / 10000 MB CPU: 0.2% / 500%

File Edit View Run Kernel Git Tabs Settings Help | Mem: 228 / 10000 MB CPU: 0.2% / 500%

salesforce_v4.ipynb

```
Filter files by name
```

... -- --

2392 46 2023

2393 45 2023

2394 46 2023

2395 46 2023

2396 46 2023

2397 rows x 8 columns

[43]:

```
# Esta función es para dividir en bloques (fragmentos) de 3500 caracteres la tabla HTML para poder ingresarla al Data # Extension de Salesforce que tiene un límite de 4000 caracteres por casilla
def armar_fragmentos(df, pid):
    df = df[df['PARTNER_ID'] == str(pid)]

    tabla_completa = build_table(df[columnas_finales],
        , 'yellow_light'
        , odd_bg_color = '#F7F7F7'
        , even_bg_color = '#FFF2F2'
        , font_size = 'medium'
        , font_family = 'Calibri'
        , text_align = 'center'
        , padding = '8px'
        , width = '100px'
        , index = False
    )

    tabla_completa = tabla_completa.replace(
        '<th>' style = "background-color: #FFFFFF;font-family: Calibri;font-size: medium;color: #BFBF00;text-align: center;border-bottom: 2px solid #BFBF00"
        , '<th>' style = "background-color: #FFCC28;font-family: Calibri;font-size: medium;color: #000000;text-align: center;border-bottom: 2px solid #FFCC28"
    )

    # Corregimos/reemplazamos los 'por' para evitar errores en el envío de datos en forma de datos extension como formato .json
    tabla_corregida = tabla_completa.replace("por", "para")
    tabla_corregida = str(tabla_corregida)

    # Fraccionamos en bloques de 3999 ya que el máximo de caracteres es de 4000 y la tabla tiene 5287.
    fragmentos = ['"', '"', '"'] # Aquí se almacenará cada elemento del HTML particionado

    niv = 1 # Se define así
```

Simple 0 1 Default_env_61503 | Idle Saving completed Mode: Command | Mem: 228 / 10000 MB CPU: 0.2% / 500%

The image shows two Jupyter Notebook sessions side-by-side, both titled "salesforce_v4.ipynb".

Top Notebook (Cell 47):

```

pivot += 1 # Se dejó así
bloques = 3500 # Se cambia a 4000, o el max length que establezcamos.

while len(tabla_corregida) > 0:
    # Guardo parte del string
    fragmentos[pivot-1] = tabla_corregida[:bloques]
    # Recorto el fragmento que acabamos de guardar
    tabla_corregida = tabla_corregida[bloques:]
    # Avanza el pivote (avanzará las veces que sea necesario en caso de necesitar más de 3, pero ya habiendo más de 3 hay que modificar el DE de SalesForce)
    pivot += 1

return fragmentos

```

Bottom Notebook (Cell 49):

```

[47]: # Obtención de bearer token
def token_bearer():
    # URL autenticación
    url = 'https://mc2tq...marketingcloudapis.com/v2/token'
    headers = {
        'Content-Type': 'application/json'
    }
    data = {
        'grant_type': 'password',
        'client_id': '...', 'client_secret': '...', 'username': '...', 'password': '...'
    }
    response = rq.request('POST', url, data=data)

    return response.json()

# Con esta función creamos el POST para enviar los datos antes de activar el entry event
def envio_correo(
    correo, # Correo del seller
    pid, # Partner_ID del seller
    seller, # Nombre del seller
    fragmentos, # Bloque 1 del HTML
    bearer_token # Bearer Token
):
    # URL del evento
    url1 = 'https://...tingcloudapis.com/interaction/v1/events'

    # Header con el bearer token
    headers2 = {
        'Content-Type': 'application/json',
        'Authorization': f'Bearer {bearer_token}'
    }

    body = {
        'ContactKey': f'{pid}',
        'EventDefinitionKey': 'APIEvent',
        'Data':
            {
                'WY_correo': f'(year_week_act){correo}',
                'Email': f'({correo})',
                'Seller': f'({seller})',
                'Tabla1': f'{fragmentos[0]}',
                'Tabla2': f'{fragmentos[1]}',
                'Tabla3': f'{fragmentos[2]}'
            }
    }

    # Hacemos el POST
    response = rq.request('POST', url1, data=json.dumps(body), headers=headers2)

    return response

# Si se ejecuta correctamente imprimirá: <Response [201]> y el eventInstanceId
# print(response)
# print(response.json())
# Response 401 es que hay que ejecutar de nuevo el bearer token o error "excede length"
# Response 201 es que fue exitoso

[49]: # Validamos los correos para enviar cada correo al respectivo Seller y eliminar los correos duplicados
def validar_correo(correo):
    patron = r'^[\w.-]+@[^\w.-]+\.\w+$'
    return bool(re.match(patron, correo))

# Creamos una lista para almacenar los correos de cada seller
lista_elementos_a_enviar = []

for row, element in df_definitivo[['PARTNER_ID', 'SELLER_NAME']].drop_duplicates().iterrows():
    lst_correos = df_correos[df_correos['PARTNER_ID'] == str(element['PARTNER_ID'])]['CORREO_FINAL'].values[0].split(';')
    lst_correos = list(set(lst_correos)) # Elimina duplicados
    for correo in lst_correos:
        lista_elementos_a_enviar.append([correo, element['SELLER_NAME']])

```

The screenshot shows two Jupyter Notebook cells running in a Python environment named 'Default_env_61503'. The top cell contains code for sending emails to partners based on specific criteria. The bottom cell contains code for creating a DataFrame from event data.

```

File Edit View Run Kernel Git Tabs Settings Help | Mem: 228 / 10000 MB CPU: 0.33% / 50
salesforce_v4.ipynb | Hi Felipe Cornejo - Vendor II Default_env_61503
+ % > Code git
for correo in list_correos:
    if validar_correo(correo) == True:
        temporal_ = {
            'WEEK': year_week_act,
            'CORREO': correo,
            'PID': elemento['PARTNER_ID'],
            'SELLER_NAME': elemento['SELLER_NAME'],
            'FRAGMENTOS': armar_fragmentos(df_definitivo, elemento['PARTNER_ID'])
        }
        lista_elementos_a_enviar.append(temporal_)

[50]: bearer_token = token_bearer()['access_token'] # Token temporal, duración 1080s
marca_tiempo = datetime.now(tz = santiago_tz) # Creamos una marca de tiempo para la duración del token
events_instances = pd.DataFrame(columns = ['EVENT_INSTANCE_ID', 'PARTNER_ID', 'SELLER_NAME', 'CORREO', 'YEAR_WEEK', 'CREATED_AT', 'CAMPAIGN'])

# Con este ciclo for enviamos todos los mails a cada partner_id con a respectivo correo
for elemento in lista_elementos_a_enviar:

    if (datetime.now(tz = santiago_tz) - marca_tiempo).seconds > 1000:
        bearer_token = token_bearer()['access_token']
        marca_tiempo = datetime.now(tz = santiago_tz)

    response = envio_correo(
        correo = elemento['CORREO'],
        pid = elemento['PID'],
        seller = elemento['SELLER_NAME'],
        fragmentos = elemento['FRAGMENTOS'],
        bearer_token = bearer_token
    )

    sleep(0.1)

    # Por si existe algún error
    try:
        if response.json()['message'].__contains__('The event data contains duplicate value for an existing primary key. Please correct the event data'):
            print("Problemas para enviar el correo del ID_STRIKE: " + elemento['PID'] + " " + elemento['CORREO'])
            continue
    except:
        event_instance_id = 'No enviado'
    try:
        print("Correo enviado ID_CORRED -> [{response.json()['eventInstanceId']}] seller: " + elemento['SELLER_NAME'])
        event_instance_id = response.json()['eventInstanceId']
    except:
        event_instance_id = 'No enviado'

Simple 0 1 Default_env_61503 | Idle Saving completed Mode: Command | Ln 13, Col 36 salesforce_v4.ipynb
File Edit View Run Kernel Git Tabs Settings Help | Mem: 228 / 10000 MB CPU: 0.24% / 500
salesforce_v4.ipynb | Hi Felipe Cornejo - Vendor II Default_env_61503
+ % > Code git
dict_ = {
    'EVENT_INSTANCE_ID': event_instance_id,
    'PARTNER_ID': elemento['PID'],
    'SELLER_NAME': elemento['SELLER_NAME'],
    'CORREO': elemento['CORREO'],
    'YEAR_WEEK': year_week_act,
    'CREATED_AT': datetime.now(tz = santiago_tz).strftime("%Y-%m-%dTH:%M:%S"),
    'CAMPAIGN': 'SP_01'
}
serie_ = pd.Series(dict_)
serie_ = serie_.to_frame()
events_instances = pd.concat([events_instances, serie_], axis = 0)

print('Correos enviados correctamente.')

```

Anexo 10: Código tasa de devolución:

The screenshot shows a Jupyter Notebook interface with two code cells. The top cell contains Python code for interacting with Salesforce using pandas and requests, and for connecting to Big Query using Google Cloud's bq library. It includes imports for pretty_html_table, pandas, numpy, requests, json, os, re, time, datetime, pytz, and google.cloud.bigquery. It defines variables like santiago_tz, iso_calendar, iso_year, iso_week, mes_actual, and key_path. It also imports credentials from a JSON file and performs Big Query queries to get seller contact information.

```
[42]: # Para mejorar la visualización de la tabla usamos la librería pretty_html_table
from pretty_html_table import build_table

# Pandas para el manejo de tablas y datos
import pandas as pd
import numpy as np

# Librerías que utilizaremos para hacer consultas hacia Salesforce
import requests as rq
import json

# Para utilizar archivos
import os

# Expresiones regulares para definir formato de casilla mail
import re

# Importamos time,sleep para el envío de mails
from time import sleep

# Definimos la timezone para obtener el dia/hora en caso de necesitarlo
from datetime import datetime, timedelta

import pytz

santiago_tz = pytz.timezone('America/Santiago') # En este caso la de Santiago, Chile
today = datetime.now(tz=santiago_tz)

iso_calendar = today.isocalendar()
iso_year = iso_calendar[0]
iso_week = iso_calendar[1]

mes_actual = today.month

# Librería para hacer querys y demas a Big Query.
from google.cloud import bigquery

# Importamos las credenciales
key_path = '/home/jupyter/Seller-Performance-Felipe/credentials.json'

# Importamos las querys de google Big Query
query_correos_path = "/home/jupyter/Seller-Performance-Felipe/Terminados/Salesforce/bq_correos_sellers.sql" # + ELEMENT_RELATIVE_PATH
query_bq_devoluciones_path = "/home/jupyter/Seller-Performance-Felipe/Terminados/Salesforce/bq_devoluciones.sql" # + ELEMENT_RELATIVE_PATH
```

The bottom cell continues the script, defining a function `df_correos_bigquery` that reads a Big Query query into a DataFrame and transforms it. It then defines another function `df_con_contactos` that reads a file, creates a DataFrame, and appends a column `CORREO_FINAL`. It lists columns in `lst_cols` and iterates through rows to update the `PARTNER_ID` column.

```
[32]: # Esta función hace una query a Big Query y devuelve un dataframe con datos de correos de sellers.

def df_correos_bigquery():

    # Leemos el contenido del archivo dado por la ruta.
    with open(query_correos_path, 'r') as file:
        query_correos = file.read()

    df_correos = client.query(query_correos).to_dataframe().astype(str)
    # Inyección de la query al cliente de Big Query y transformación a DataFrame.

    return df_correos

[33]: # Esta función nos permite obtener todos los mail de contacto de los sellers.

def df_con_contactos():
    df_correos = df_correos_bigquery()
    df_correos['CORREO_FINAL'] = None
    # Armamos un DataFrame con los correos que se pueden encontrar en Salesforce.

    lst_cols = [
        'Correo_Representante_Legal',
        'Correo_Contacto_Comercial',
        'Correo_Contacto_Logistica',
        'Correo_Contacto_SAC',
        'Correo_Contacto_Pagos',
        'Correo_Contacto_Capacitaciones',
        'Correo_Contacto_Performance',
        'Correo_Contacto_Otro'
    ]
    # List con correos a utilizar.

    for row, _ in enumerate(df_correos['PARTNER_ID']):
        if row % 100 == 0:
            print(f'Procesando fila {row} de {len(df_correos)}')

        partner_id = df_correos['PARTNER_ID'][row]
```

File Edit View Run Kernel Git Tabs Settings Help | Mem: 392 / 10000 MB CPU: 0.23% / 500%

Filter files by name

- / Seller-Performance-Felipe /
 - Name Last Modified
 - En progreso 6 days ago
 - Terminados 29 minutes ago
 - credentials... a month ago
 - README.md a month ago
 - salesforce... a minute ago

```

for row_ in enumerate(df_correos['PARTNER_ID']):
    correo_final = ""

    if df_correos.at[row_, 'Correo_Contacto_Performance'] is 'No Registrado':
        correo_final = df_correos.at[row_, 'Correo_Contacto_Performance'] # Privilegiamos el correo de "performance"

    else:
        # Si no hay correo_contacto_performance, concatenamos los que encontramos.
        for col in lst_cols:
            if df_correos.at[row_, col] != 'No Registrado':
                correo_final += df_correos.at[row_, col] # Concatenamos los correos encontrados

    if len(correo_final) > 0:
        df_correos.at[row_, 'CORREO_FINAL'] = correo_final # Concatenación de todos los correos

    else:
        df_correos.at[row_, 'CORREO_FINAL'] = "Sin correo registrado." # Valor por defecto

return df_correos[['PARTNER_ID', 'CORREO_FINAL']]

```

[43]: # Esta función nos permite obtener los datos de las columnas que tendrá la tabla que se muestra en el correo.
def bbdd_resumen_ordenes_mensual():

```

# Leemos el contenido del archivo
with open(query_bbdd_devoluciones_path, 'r') as file:
    query_ordenes = file.read()

dict_valores = {
    'YEAR': np.int16,
    'MONTH': np.int16,
    'PARTNER_ID': np.int64,
    'SELLER_NAME': str,
    'LINEAS_TOTALES': np.int16,
    'ORDENES_DEVUeltas': np.int16,
    'TASA_DEVOLUCION': np.float32
}

# Proceso de ejecución en BigQuery
df_ordenes = client.query(query_ordenes).to_dataframe(dtypes = dict_valores)

return df_ordenes

```

Simple 0 2 No Kernel | Idle Saving completed Mode: Edit L 19, Col 81 salesforce_devoluciones.ipynb

File Edit View Run Kernel Git Tabs Settings Help | Mem: 392 / 10000 MB CPU: 0.23% / 500%

Filter files by name

- / Seller-Performance-Felipe /
 - Name Last Modified
 - En progreso 6 days ago
 - Terminados 29 minutes ago
 - credentials... a month ago
 - README.md a month ago
 - salesforce... 2 minutes ago

```

[44]: # Obtenemos los datos de venta
df_definitivo = bbdd_resumen_ordenes_mensual()

# df con los correos de los sellers
df_correos = df_com_correos()

df_definitivo

```

	YEAR	MONTH	PARTNER_ID	SELLER_NAME	LINEAS_TOTALES	ORDENES_DEVUeltas	TASA_DEVOLUCION
0	2023	11			71	2	2.82
1	2023	11			1	0	0.00
2	2023	11			11	1	9.09
3	2023	11			57	0	0.00
4	2023	11			372	3	0.81
...	—	—
1040	2023	10			1	0	0.00
1041	2023	10			1	0	0.00
1042	2023	10			1	0	0.00
1043	2023	10			1	0	0.00
1044	2023	10			1	0	0.00

1045 rows x 7 columns

```

[45]: # Acá se comienza a filtrar la tabla

# Agregamos el símbolo de % y corregimos decimales en 0% y 100%
df_definitivo['TASA_DEVOLUCION'] = df_definitivo['TASA_DEVOLUCION'].astype(float)
df_definitivo['TASA_DEVOLUCION'] = df_definitivo['TASA_DEVOLUCION'].map('{:.1f}%'.format)
df_definitivo['TASA_DEVOLUCION'] = df_definitivo['TASA_DEVOLUCION'].replace(to_replace = '0.0%', value = '0%')
df_definitivo['TASA_DEVOLUCION'] = df_definitivo['TASA_DEVOLUCION'].replace(to_replace = '100.0%', value = '100%')

dict_rename = {
    'YEAR': 'Año',
    'MONTH': 'Mes',
}

```

Simple 0 2 No Kernel | Idle Saving completed Mode: Edit L 19, Col 81 salesforce_devoluciones.ipynb

File Edit View Run Kernel Git Tabs Settings Help | Mem: 393 / 10000 MB CPU: 0.23% / 500%

salesforce_devoluciones.ipynb X

```

df_definitivo['TASA_DEVOLUCION'] = df_definitivo['TASA_DEVOLUCION'].replace(to_replace = '100.0%', value = '100')

dict_rename = {
    'YEAR': 'Año',
    'MONTH': 'Mes',
    'PARTNER_ID': 'PARTNER_ID',
    'SELLER_NAME': 'SELLER_NAME',
    'LINEAS_TOTALES': 'Líneas totales',
    'ORDENES_DEVUELVTAS': 'Líneas devueltas',
    'TASA_DEVOLUCION': 'Tasa de devolución (%)'
}

# Renombramos las columnas para que sea más amigable para el seller
df_definitivo = df_definitivo.rename(dict_rename, axis = 1)

# Recortamos el df
df_definitivo = df_definitivo[list(dict_rename.values())]

# Convertimos el df a string para poder utilizarlo en el ciclo for
df_definitivo = df_definitivo.astype(str).sort_values(by = ['PARTNER_ID', 'Mes', 'Año'], ascending = [1, 0, 0]).reset_index(drop = True)

columnas_finales = [
    'Año',
    'Mes',
    'Líneas totales',
    'Líneas devueltas',
    'Tasa de devolución (%)'
]
df_definitivo

```

	Año	Mes	PARTNER_ID	SELLER_NAME	Líneas totales	Líneas devueltas	Tasa de devolución (%)
0	2023	11			15	0	0%
1	2023	11			8	0	0%
2	2023	11			572	3	0.5%
3	2023	11			129	2	1.6%
4	2023	11			454	6	1.3%
...
725	2022	11			6	0	0%
726	2022	11			11	0	0%

Simple 0 2 No Kernel | Idle Saving completed Mode: Edit L 19, Col 81 salesforce_devoluciones.ipynb

File Edit View Run Kernel Git Tabs Settings Help | Mem: 392 / 10000 MB CPU: 1.22% / 500%

salesforce_devoluciones.ipynb X bq_devoluciones.sql X

```

[47]: # Esta función es para dividir en bloques (fragmentos) de 3500 caracteres la tabla HTML para poder ingresarla al Data
# Extension de Salesforce que tiene un límite de 4000 caracteres por casilla
def armar_fragmentos(df, pid):
    df['PARTNER_ID'] == str(pid)

    tabla_completa = build_table(df[columnas_finales]
        , yellow_light
        , odd_bg_color = '#F7F7F7'
        , even_bg_color = '#FFF2F2'
        , font_size = 'medium'
        , font_family = 'Calibri'
        , text_align = 'center'
        , padding = '5px'
        , width = '100px'
        , index = False
    )

    tabla_completa = tabla_completa.replace(
        '<th style = "background-color: #FFFFFF;font-family: Calibri;font-size: medium;color: #BFBFBF;text-align: center; border-bottom: 2px solid #BFBFBF;>'
        , '<th style = "background-color: #FFFFFF;font-family: Calibri;font-size: medium;color: #BFBFBF;text-align: center; border-bottom: 2px solid #FFFFFF;>'
    )
    # Corregimos/reemplazamos los ' ' por '' para evitar errores en el envío de datos en forma de datos extension como formato .json
    tabla_corregida = tabla_completa.replace(' ', '')
    tabla_corregida = str(tabla_corregida)

    # Fraccionamos en bloques de 3999 ya que el máximo de caracteres es de 4000
    fragmentos = []
    # Aquí se almacenará cada elemento del HTML particionado
    bloques = 3500 # Se cambia a 4000, o el max_length que establezcamos.

    pivot = 1 # Se deja así
    bloques = 3500 # Aquí se almacenará cada elemento del HTML particionado

    while len(tabla_corregida) > 0:
        # Guardo parte del string
        fragmentos.append(tabla_corregida[:bloques])
        # Recorto el fragmento que acabamos de guardar
        tabla_corregida = tabla_corregida[bloques:]
        # Avanza el cursor
        pivot += 1

    return fragmentos

```

Simple 0 2 No Kernel | Idle Saving completed Mode: Edit L 27, Col 80 salesforce_devoluciones.ipynb

```

[saleforce_devoluciones.ipynb]: bq_devoluciones.sql
[47]: 
def token_bearer():
    # URL autenticación
    url = "https://"
    headers = {
        'Content-Type': 'application/json'
    }
    data = {
        'grant_type': 'client_credentials',
        'client_id': 'XXXXXXXXXX',
        'client_secret': 'XXXXXXXXXX'
    }
    response = rq.request('POST', url=url, data=data)

    return response.json()

# Con esta función creamos el POST para enviar los datos antes de activar el entry event
def envio_correo(
    correo,          # Correo del seller
    pid,             # Partner_ID del seller
    seller,          # Nombre del seller
    fragmentos,     # Bloque 1 del HTML
    bearer_token    # Bearer Token
):
    # URL del evento
    url2 = 'https://mi'

    # Header con el bearer token
    headers2 = [
        'Content-Type': 'application/json',
        'charset' : 'UTF-8',
        'Authorization': f'Bearer {bearer_token}'
    ]
    # Body del POST
    body = {
        'contactKey': f'{pid}',
        'eventDefinitionKey': '',
        'Data': [
            {
                'YM_correo': f'{mes_actual}{correo}',
                'Email': f'{correo}',
                'Seller': f'{seller}',
                'Table1': f'{fragmentos[0]}',
                'Table2': f'{fragmentos[1]}'
            }
        ]
    }

    # Hacemos el POST
    response = rq.request('POST', url2, data=json.dumps(body), headers=headers2)

    return response

# Si se ejecuta correctamente imprimirá: <Response [201]> y el eventInstanceId
# print(response)
# print(response.json())
# Response 401 es que hay que ejecutar de nuevo el bearer token o error "excede Length"
# Response 200 es que fue exitoso

[49]: #Validamos los correos para enviar cada correo al respectivo Seller y eliminar los correos duplicados
def validar_correo(correo):
    patron = r"[\w\.-]+@[^\.\-]+\.\w+"
    return bool(re.match(patron, correo))
# Creamos una lista para almacenar los correos de cada seller
lista_elementos_a_enviar = []

for row, element in df_definitivo[['PARTNER_ID', 'SELLER_NAME']].drop_duplicates().iterrows():
    lst_correos = df.correos[df.correos['PARTNER_ID'] == str(element['PARTNER_ID'])]['CORREO_FINAL'].values[0].split(';')
    lst_correos = list(set(lst_correos)) # Elimina duplicados
    for correo in lst_correos:
        if validar_correo(correo) == True:
            temporal_ = {
                '#_WY': year_week_act,
                'CORREO': correo,
                'PID': element['PARTNER_ID'],
                'SELLER_NAME': element['SELLER_NAME'],
                'FRAGMENTOS': armar_fragmentos(df_definitivo, element['PARTNER_ID'])
            }
            lista_elementos_a_enviar.append(temporal_)

[50]: bearer_token = token_bearer()['access_token'] # Token temporal, duración 1080s
marca_tiempo = datetime.now(tz = santiago_tz) # Creamos una marca de tiempo para la duración del token
events_instances = pd.DataFrame(columns = ['EVENT_INSTANCE_ID', 'PARTNER_ID', 'SELLER_NAME', 'CORREO', 'MONTH', 'CREATED_AT', 'CAMPAIGN'])

```

```

File Edit View Run Kernel Git Tabs Settings Help | Mem: 392 / 10000 MB CPU: 0.22% / 500%
+ C gt
salesforce_devoluciones.ipynb bq_devoluciones.sql
Filter files by name
Name Last Modified
En progreso 6 days ago
Terminados 31 minutes ago
credentials... a month ago
README.md a month ago
salesforce... a minute ago
for elemento in lista_elementos_a_enviar:
    if (datetime.now(tz = santiago_tz) - marca_tiempo).seconds > 1000:
        bearer_token = token_bearer()['access_token']
        marca_tiempo = datetime.now(tz = santiago_tz)

        response = envio_correo(
            correo = elemento['CORREO'],
            pid = elemento['PID'],
            seller = elemento['SELLER_NAME'],
            fragmentos = elemento['FRAGMENTOS'],
            bearer_token = bearer_token
        )

        sleep(0.1)

    # Por si existe algún error
    try:
        if response.json()['message'].contains_('The event data contains duplicate value for an existing primary key. Please correct the event data.')
            print("Problemas para enviar el correo del ID_STRIKE: " + str(elemento['PID']) + str(elemento['CORREO']))
            continue
    except:
        event_instance_id = 'No enviado'

    try:
        print("Correo enviado ID_CORREO -> [" + str(response.json()['eventInstanceId']) + "] seller: " + str(elemento['SELLER_NAME']))
        event_instance_id = response.json()['eventInstanceId']
    except:
        event_instance_id = 'No enviado'

    dict_ = {
        'EVENT_INSTANCE_ID': event_instance_id,
        'PARTNER_ID': elemento['PID'],
        'SELLER_NAME': elemento['SELLER_NAME'],
        'CORREO': elemento['CORREO'],
        'MONTH': mes_actual,
        'CREATED_AT': datetime.now(tz = santiago_tz).strftime("%Y-%m-%d %H:%M:%S"),
        'CAMPAIGN': 'SP_01'
    }
    serie_ = pd.Series(dict_)
    serie_ = serie_.to_frame().T
    events_instances = pd.concat([events_instances, serie_], axis = 0)
print('Correos enviados correctamente.')

```