UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

CURSO DE ESPECIALIZACIÓN EN INGENIERÍA DE DATOS

PROPUESTA DE ARQUITECTURA DE DATA WAREHOUSE PARA EL ANÁLISIS, PROCESAMIENTO Y VISUALIZACIÓN DE DATOS DEL ÁREA DE VENTAS DE LA DISTRIBUIDORA EL ÁGUILA

PRESENTADO POR:

JOSUE ALFONSO TORRES ARTIAGA

RODRIGO MAURICIO TORRES ARTIAGA

PARA OPTAR AL TÍTULO DE:

INGENIERO DE SISTEMAS INFORMÁTICOS

UNIVERSIDAD DE EL SALVADOR

CIUDAD UNIVERSITARIA, ENERO DE 2023

RECTOR:

MSC. ROGER ARMANDO ARIAS ALVARADO

SECRETARIO GENERAL:

MSC. FRANCISCO ANTONIO ALARCÓN SANDOVAL

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO:

PHD. EDGAR ARMANDO PEÑA FIGUEROA

SECRETARIO:

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

DIRECTOR:

ING. RUDY WILFREDO CHICAS VILLEGAS

UNIVERSIDAD DE EL SALVADOR

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO DE SISTEMAS INFORMÁTICOS

Título:

PROPUESTA DE ARQUITECTURA DE DATA WAREHOUSE PARA EL ANÁLISIS, PROCESAMIENTO Y VISUALIZACIÓN DE DATOS DEL ÁREA DE VENTAS DE LA DISTRIBUIDORA EL ÁGUILA

Presentado por:

JOSUE ALFONSO TORRES ARTIAGA

RODRIGO MAURICIO TORRES ARTIAGA

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. RENÉ FABRICIO QUINTANILLA

SAN SALVADOR, ENERO DE 2023

Trabajo de Graduación Aprobado por:

Docente Asesor:

MSc. René Quintanilla

Índice

Contenido

Agradecimientos

Agradezco a mi madre, quien fue la persona que más apoyo me dio para lograr esta meta que me propuse desde que tenía 15 años, a mi hermano con quien trabaje siempre lado a lado luchando por lograr el mismo objetivo y que ahora podemos compartir la misma alegría, y a mi padre por apoyarme.

Doy gracias a todos y cada uno de los compañeros y amigos que conocí en el transcurso de mi formación, porque aprendí mucho de ellos, así como probablemente ellos de mí.

Agradezco igualmente a los profesores que tuve a lo largo de mi formación, por ser lo más estrictos y duros al evaluar mi potencial.

Finalmente, me agradezco a mí, por no rendirme nunca pese a todo lo vivido, todo valió la pena.

Josue Alfonso Torres Artiaga

Agradezco a mi mamá por haberme apoyado en todo momento, por estar pendiente de mí, y por ser una de las personas más importantes en mi vida. Sin ella, no hubiese podido cumplir este logro.

Agradezco a mi hermano, Josue, por estar siempre a mi lado y por escucharme. Me ayudó y apoyó cuando lo necesitaba.

Agradezco a mi papá por haberme ayudado en lo que estuvo a su alcance, por los consejos y por estar pendiente de mí.

Agradezco a Dios que hizo posible el poder alcanzar este logro. Me brindó paciencia, perseverancia, valor, y a mi familia y amigos.

Agradezco a mis mascotas porque siempre estaban conmigo. Tanto a mis perritos como gatitos, a los que aún están y a los que no.

Me agradezco a mí mismo por no haberme rendido, aunque el camino fuese difícil.

Rodrigo Mauricio Torres Artiaga

Introducción

En el presente documento se detalla cómo se analizó, diseñó y propuso una solución para el problema identificado en la empresa Distribuidora El Águila. El documento se divide en tres capítulos: Especificación del proyecto, Análisis y diseño de la propuesta de solución y Estrategia de implementación de la propuesta de solución.

En el primer capítulo, se describe la situación actual de la empresa. Incluyendo sus antecedentes, en los que se define si ya contaban con una solución que les ayudará a tomar decisiones basadas en información o no. Además, se describe el problema que será solucionado por la propuesta de solución. El problema se analiza utilizando la técnica de la caja negra y el enfoque de sistemas. Esto con el objetivo de identificar, definir y plantear correctamente el problema que se busca solucionar. Por otro lado, también se definen los objetivos que trazarán las acciones y decisiones a tomar en este proyecto. A continuación, se presentan los alcances del proyecto que definen hasta dónde llegará este último. Continuando, se presenta la justificación que plantea por qué se decidió realizar este proyecto. Para concluir con el primer capítulo, se presenta el cronograma de actividades y el presupuesto necesario en la etapa de desarrollo de la propuesta de solución.

En el segundo capítulo, se presenta la metodología de trabajo que se sigue para el desarrollo del proyecto. Dicha metodología desglosa el análisis de la arquitectura de data warehouse a utilizar, el análisis de los sistemas fuentes y de las métricas, el procesamiento de los datos (Extracción, Transformación y Carga de Datos), el área de presentación de datos y las aplicaciones de BI a utilizar. Además, se presenta la descripción de las tecnologías a utilizar, en donde se detallan las tecnologías utilizadas para desarrollar la propuesta de solución. Después, se muestra un diagrama arquitectónico de la solución

que muestra los componentes arquitectónicos de la misma. Para concluir este capítulo, se describe de qué forma fueron utilizadas las tecnologías descritas en el punto anterior.

En el tercer capítulo, se presenta una estrategia de implementación de la solución propuesta, en donde se propone cómo se podría llevar a cabo la solución. Continuando con esta estrategia, se presentan los posibles costos a los que tendría que incurrir la empresa para poder implementar la propuesta de solución. Para finalizar, se muestra un análisis de los resultados obtenidos mediante la implementación de la propuesta.

Para finalizar, se presentan las conclusiones basadas en los objetivos planteados y las recomendaciones propuestas para la operación, implementación y mantenimiento de la propuesta de solución.

Situación actual

Antecedentes

La empresa Distribuidora El Águila es una empresa comercial con su rubro en el área de electrodomésticos (productos de línea blanca). El nacimiento de la empresa se da a inicios de 2022 como una iniciativa salvadoreña. Actualmente, la empresa posee solamente una sucursal ubicada en San Salvador. Además de tener una tienda en línea implementada con el CMS (Content Management System) de PrestaShop.

La empresa tiene definida su misión, visión y valores:

Misión

Crear una opción rentable que ayudará a la población salvadoreña a conseguir los electrodomésticos más confiable y tecnológicamente actualizados que les permita gozar de una mejor comodidad.

Visión

Ser un ejemplo de compromiso con el cliente.

Valores

- Responsabilidad
- Transparencia
- Integridad
- Respeto
- Compromiso

La estructura organizativa de la empresa es la siguiente:

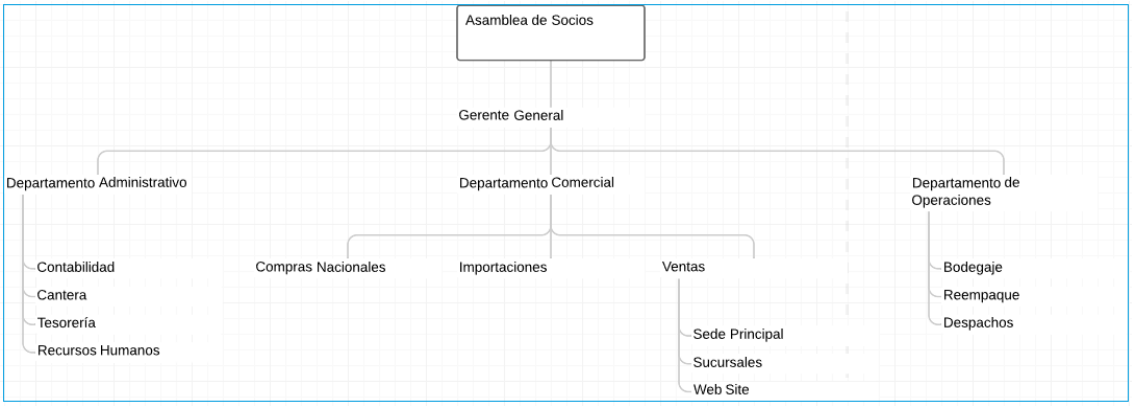

Figura 1: Organigrama de Distribuidora El Águila

A continuación, se detalla la unidad a la que se le brindó apoyo:

- Web Site (Ventas): Esta unidad se encarga de brindar apoyo al departamento de ventas para lograr alcanzar las metas propuestas, implementar las estrategias, mejorar la atención al cliente y promocionar a la empresa. Todo esto gracias al uso de la tecnología y a las estrategias comerciales planteadas por el departamento de ventas.

Actualmente, la empresa no cuenta con ninguna forma que le permita analizar los datos generados por todas las ventas que se realizan, ya sean ventas desde la sucursal física o desde el sitio web. La empresa necesita de una solución que les permita analizar los datos generados de manera que se puedan crear o mejorar las estrategias de ventas y generar más ingresos mientras se minimizan los costos para la empresa.

Los administradores ejecutan y basan sus estrategias en un método empírico desarrollado con base en la experiencia, pero este último no es muy confiable y en ocasiones se generan muchos costos de inventario debido a que se ordenan muchos productos que al final no se venden.

Muchos de estos gastos podrían reducirse significativamente si los administradores y el gerente de ventas pudiesen tomar decisiones basados en información. Con dicha información se podrían determinar los productos con mayor rotación en el inventario y así poder reducir el costo de los mismos.

Descripción del problema

La empresa Distribuidora El Águila vende productos de línea blanca desde su sitio web y desde su sucursal física. Debido a que las ventas se pueden realizar por medio de ambos medios o canales, la distribuidora presenta una estrategia de marketing llamada omnicanalidad. La empresa tiene la necesidad de saber por qué medio se realizan más ventas, además de otra información relevante. De esa forma, la empresa podría establecer objetivos y metas claras y realistas que colaboren con el crecimiento de la misma, desarrollar mejores estrategias de ventas para lograr dichos objetivos, mejorar la atención al cliente basándose en información valiosa para la empresa y que garantice la satisfacción de estos, promocionar a la empresa mediante publicidad, entre otros beneficios.

Debido a que la empresa se dedica a la venta de productos de línea blanca, esta considera de suma importancia la necesidad de tomar decisiones basadas en información útil, sin redundancia, y confiable. La empresa necesita saber qué productos se venden más, así podría ajustar sus estrategias de ventas para solicitar más de esos productos.

De igual forma, la empresa se podría beneficiar sabiendo qué productos se venden menos, así podría solicitar menos de ese producto a sus proveedores y ahorraría más en inventario. Los productos que ocupan más espacio y que menos se venden, son un costo que podría reducirse conociendo dicha información.

En ese sentido, la empresa pretende reducir muchos costos mediante la toma de decisiones basadas en información.

Planteamiento del problema

Para plantear, analizar y definir el problema, se utilizarán los diagramas de caja negra y el enfoque de sistemas.

A continuación, se presenta el diagrama de caja negra en el que se puede observar el estado actual de la empresa Distribuidora El Águila, en términos de las bases con que se basan para tomar decisiones, y el estado que se pretende alcanzar con la arquitectura de data warehouse propuesta.

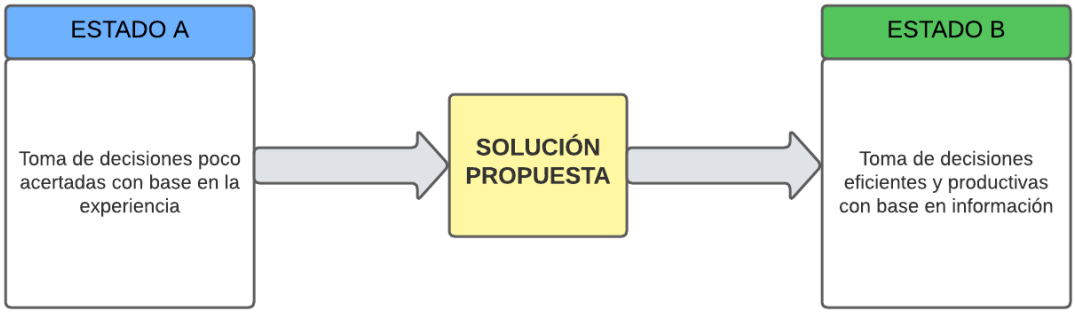

Figura 2: Diagrama de Caja Negra que muestra la situación actual de la empresa (Estado A) y la situación con la propuesta de solución (Estado B)

Continuando con el análisis del problema, también se hizo uso del enfoque de sistemas, que se puede ver a continuación:

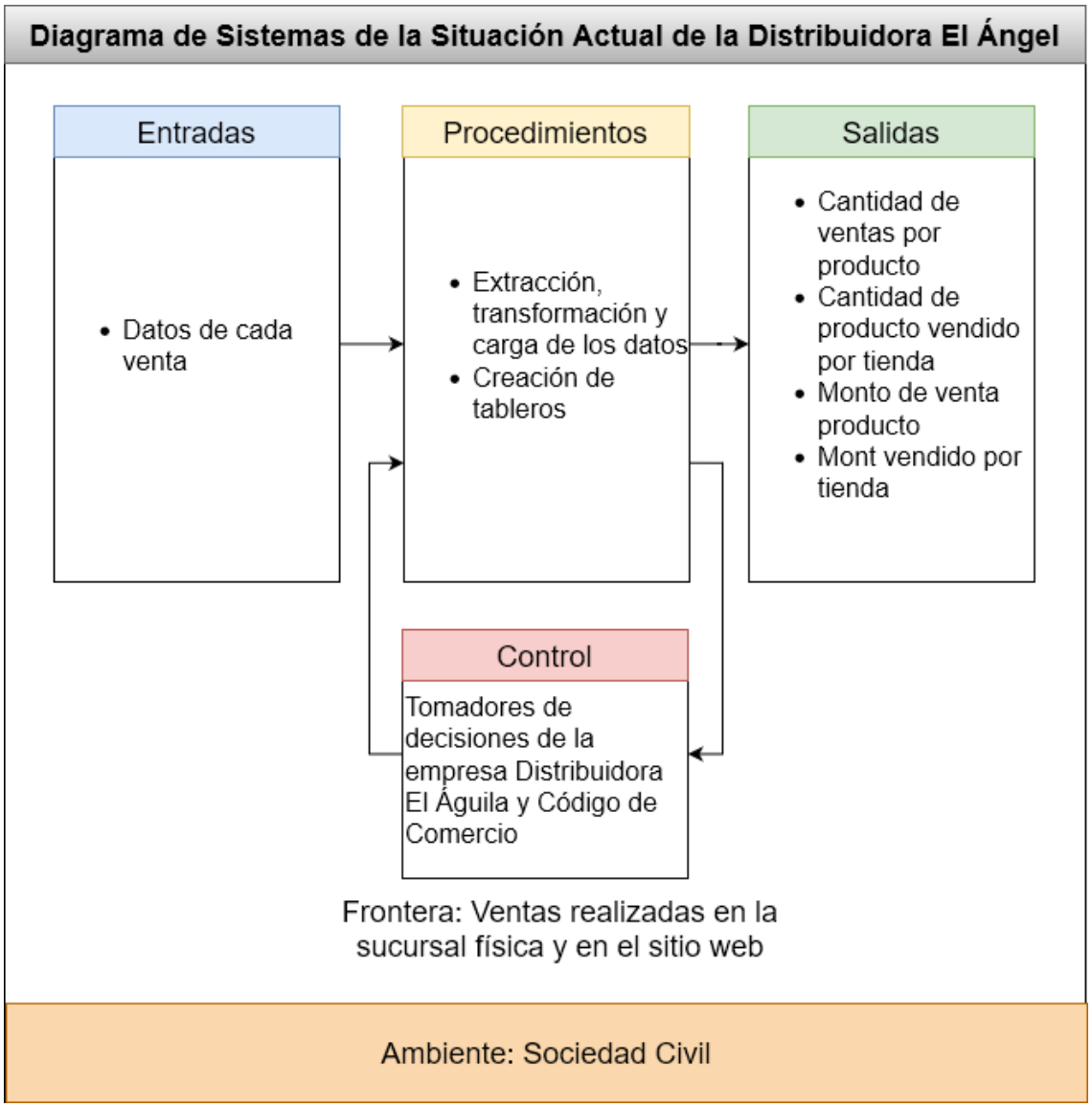

Figura 3: Enfoque de sistemas

Con base a la descripción, definición y análisis realizado, se concluye que la distribuidora El Águila, específicamente el departamento de ventas, presenta un problema a la hora de tomar decisiones gerenciales. Por tanto, se propone una arquitectura de data warehouse que permita el almacenamiento, procesamiento y presentación de información que sea útil para tomar decisiones eficientes y útiles para la empresa.

Objetivos

Objetivo general

Investigar y analizar un modelo de arquitectura de data warehouse para la implementación de una propuesta de solución al análisis de datos del área de ventas para la empresa Distribuidora El Águila que facilite la toma de decisiones basadas en información.

Objetivos específicos

- Analizar la situación actual de la empresa Distribuidora El Águila para el desarrollo de la propuesta de solución
- Analizar los esquemas de base de datos actuales para comprobar y garantizar la respuesta a las métricas planteadas
- Diseñar la propuesta de arquitectura de data warehouse
- Construir la propuesta de solución garantizando la calidad y presentación de los datos
- Diseñar un plan de implementación de la propuesta de solución

Alcances

Al haber concluido el proyecto, se presentará la propuesta de una arquitectura de data warehouse que le permita al departamento de ventas tomar decisiones más acertadas y basadas en información. Se pretende entregar la arquitectura de data warehouse que comprende el almacenamiento de los datos, su procesamiento que se compone de la transformación y carga de los mismos, y la presentación de dichos información en la aplicación llamada Microsoft Power BI.

El sistema a presentar permitirá visualizar reportes en tableros organizados. La información presentada en dichos tableros permitirá a los usuarios visualizar ciertas métricas o información relevante al negocio, que les facilite la toma de decisiones. Cabe recalcar que esta propuesta de solución será solamente una herramienta para los tomadores de decisiones.

Además, se presentará una estrategia de implementación para dicha propuesta de solución.

Justificación

A la luz de la situación actual de la distribuidora «El Águila», esta propuesta de solución es importante, dado que la implementación de dicha propuesta, podría beneficiar en gran medida a la empresa. El principal beneficio que se obtendría es que los tomadores de decisiones tendrían una herramienta en la cual apoyarse para decidir eficientemente. De esto se derivarían más beneficios: Establecer objetivos claros y realistas, desarrollar o mejorar estrategias de ventas para lograr los objetivos, mejorar la atención al cliente, y promocionar la empresa.

Por tanto, se pretende beneficiar a la distribuidora «El Águila» con la implementación de una arquitectura de data warehouse que provea de información útil a los tomadores de decisiones del departamento de Ventas para mejorar la situación actual de la empresa.

Cronograma de actividades

A continuación, se presentan las tareas ejecutadas en las dos etapas que comprenden el proyecto:

Nombre	Fecha de inicio	Fecha de fin
Primera Etapa	27/05/22	30/06/22
Introducción a la lógica del negocio	27/05/22	2/06/22
Descripción del dataset y diccionario de datos del dataset	3/06/22	9/06/22
Resultado del data profiling	10/06/22	13/06/22
Especificación de necesidades analíticas que el modelo propuesto solventará	14/06/22	17/06/22
Modelo dimensional propuesto	20/06/22	28/06/22
Mappings por tabla	29/06/22	30/06/22

Figura 4: Tareas que comprendían la Primera Etapa

Segunda Etapa	4/07/22	6/12/22
Refinamiento del análisis dimensional propuesto en la Primera Etapa	4/07/22	12/07/22
Carga de datos en el sistema origen	13/07/22	21/07/22
Configuración del entorno S3 en Amazon AWS	25/07/22	2/08/22
Configuración de Amazon IAM	3/08/22	11/08/22
Creación de base de datos en Amazon Redshift	15/08/22	23/08/22
Procesamiento de datos en Talend	24/08/22	15/11/22
Creación de tableros en Power BI	16/11/22	6/12/22

Figura 5: Tareas que comprendían la Segunda Etapa

A continuación, se presenta el gráfico del cronograma de actividades:

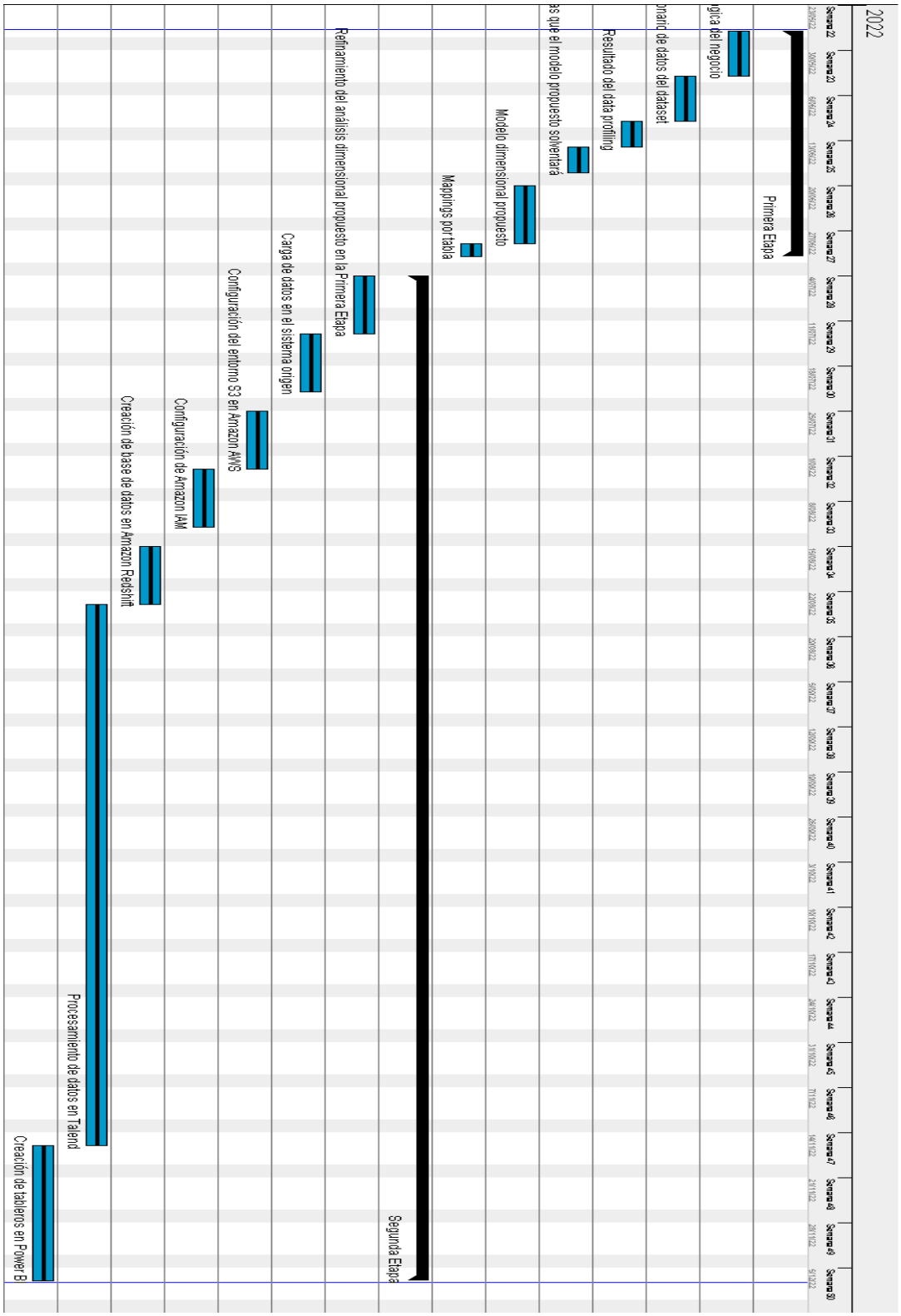

Figura 6: Gráfico del cronograma de actividades

Presupuesto

En esta sección se presentan los costos incurridos durante el desarrollo de la propuesta de solución. Se consideran todos los recursos que fueron necesarios para el desarrollo del proyecto. Se utilizó recurso humano, tecnológico y servicios básicos.

Recurso humano

Tomando como base el salario de un ingeniero de datos ($1300):

Recurso humano	**Cantidad**	**Salario mensual ($)**	**Total ($)**
Ingeniero de datos	2	1300	1300
Total			2600

Recurso tecnológico

Para determinar el monto de la depreciación del equipo, se utiliza el Método de Línea Recta.

El porcentaje se calcula usando la siguiente fórmula:

$$Porcentaje\ de\ depreciación = \left(\frac{\left(\frac{Costo\ inicial}{n\ años\ de\ vida\ útil} \right)}{12\ meses} \right) * 7\ meses\ de\ desarrollo$$

Figura 7: Método de Línea Recta

Equipo	**Porcentaje de depreciación**	**Años vida útil**	**Costo inicial ($)**	**Monto de la depreciación ($)**
Computadora 1	0.15	5	1200	140
Computadora 2	0.15	5	1200	140
Depreciación en 7 meses de				280

desarrollo del proyecto				

Servicios básicos

Recurso	**Tiempo estimado (meses)**	**Monto mensual ($)**	**Total ($)**
Energía eléctrica	7	25	175
Agua	7	10	70
Teléfono e Internet	7	40	280
Total			525

Costo total del desarrollo

Considerando que el desarrollo de la propuesta de solución es de 7 meses, se procede a multiplicar los costos mensuales por la cantidad de meses del desarrollo. Con esto se obtiene el costo total del desarrollo de la propuesta de solución.

Recurso	**Total ($) en 7 meses**
Humano	18200
Tecnológico	280
Servicios básicos	525
Total	19005

Metodología de trabajo

A continuación, se describe en los siguientes puntos cómo se abordarán los requerimientos definidos:

- **Análisis de la arquitectura de Data Warehouse a utilizar**: En este punto en concreto, se realizará un análisis de la arquitectura de data warehouse a emplear y de esta manera definir cómo será el flujo de procesamiento de los datos para su puesta en la capa de presentación a los usuarios del mismo.
- **Análisis de los sistemas fuentes y de las métricas**: En esta etapa se hará una revisión de los esquemas de base de datos de los sistemas fuentes. En este caso de la base de datos a la que se conecta el CMS PrestaShop; esto con el objetivo de entender cómo están relacionados los datos, qué datos son los que se almacenan, cuál es la calidad de los datos almacenados y qué estructura tienen definidas las tablas de base de datos. Todo esto con el objetivo de definir si es posible contestar a las preguntas de negocio (métricas), por lo que se concluirá en esta etapa luego del análisis de los sistemas fuentes, qué métricas puede contestarse y cuáles no. Además de definir el esquema y tablas de base de datos a procesar y definir a alto nivel el modelo dimensional a utilizar.
- **Procesamiento de datos (Extracción, Transformación y Carga de datos)**: Para el procesamiento de datos en esta etapa, ya se tiene definido cuál será el modelo dimensional a usar. Se define de qué manera se extraerán los datos, cómo se transformarán, para finalmente ser cargados en el esquema de estrella diseñado.
- **Área de presentación de datos**: En este punto se trabajará en construir la estructura de tablas del esquema de estrella a ser leído por las aplicaciones que mostrarán datos al usuario.
- **Aplicaciones de BI**: Finalmente se trabajará en la construcción de tableros o reportes para la presentación de datos al usuario, mediante la lectura del esquema construido en el punto anterior.

Descripción de la propuesta de solución

A continuación, se describe de manera detallada cómo se llevó a cabo cada uno de los puntos anteriormente expuestos en el apartado de Metodología de Trabajo.

Análisis de la Arquitectura de Data Warehouse a utilizar

Arquitecturas de Data Warehouse

- **Arquitectura de Kimball**

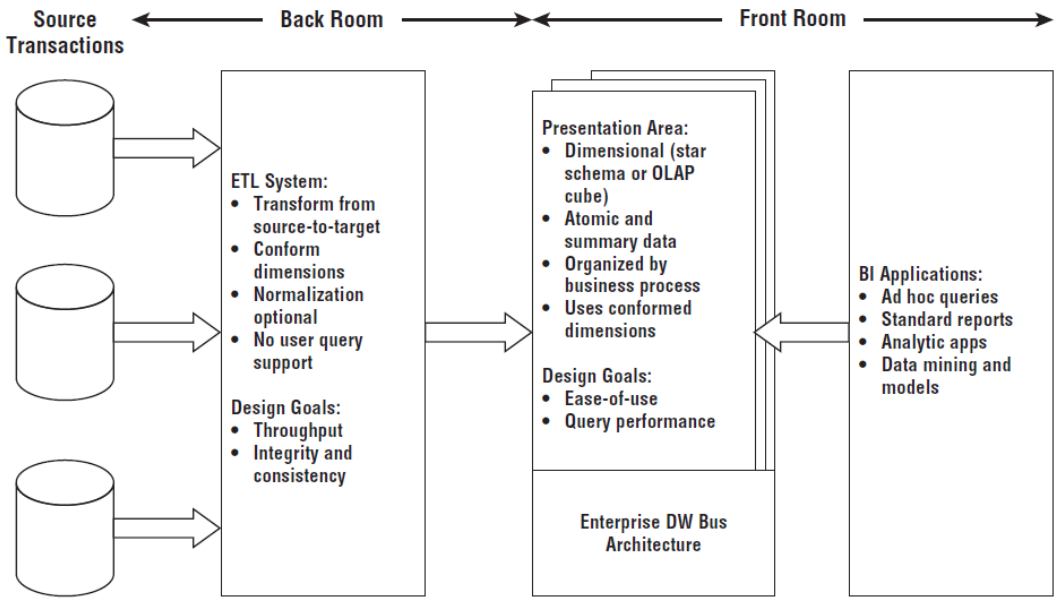

Figura 8: Arquitectura de Kimball

La arquitectura de Kimball consiste de tres áreas principales: Sistemas origen o fuente (Source transactions), Back room (también conocido como back office) y Front room o área de presentación (también conocida como Arquitectura de Bus empresarial).

Los sistemas fuentes, es el origen de donde se obtiene los datos en estado crudo, ejemplo: un sistema de ventas.

El back room, es un conjunto de procesos ETL que se encarga de la extracción, limpieza y procesamiento, hasta su carga en un repositorio de datos o data lake.

El front room, consiste de un esquema dimensional que tiene los datos en calidad para ser consumidos por aplicaciones de BI (Business Intelligence).

- **Arquitectura de Inmon**

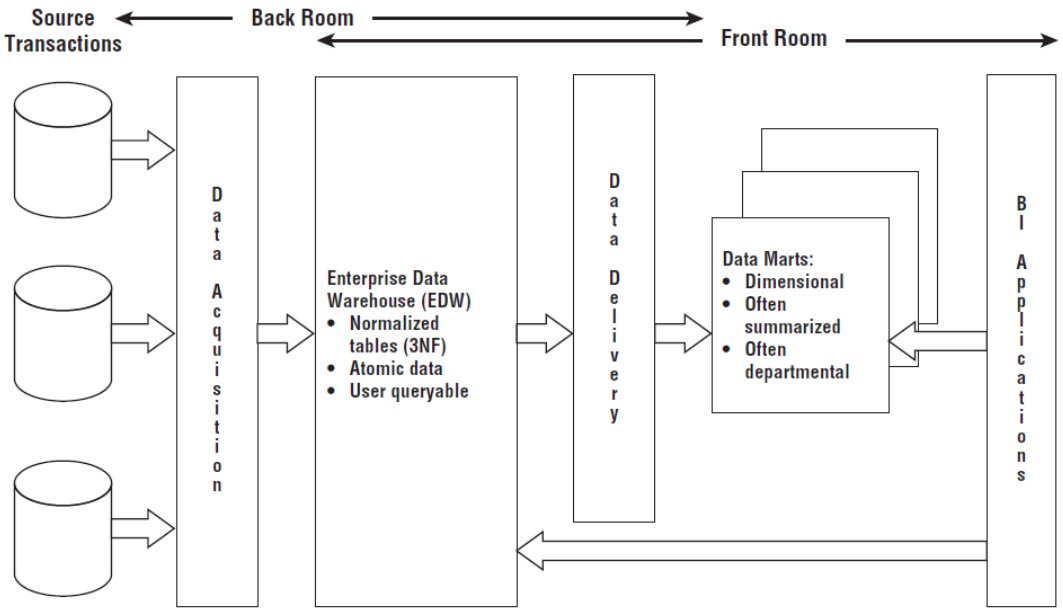

Figura 9: Arquitectura de Inmon

Esta arquitectura es ampliamente utilizada para la construcción de data lakes, en esta arquitectura se plantea que en el back room hay ciertos procesos de adquisición de dato. De modo que luego de esa etapa de adquisición de datos, estos datos se pasan a una data warehouse empresarial donde los datos son almacenados en tablas normalizada, de forma atómica, ósea con gran nivel de detalle y proveen una manera de hacer consultas. El componente de Data Delivery, es la forma en la que se extraen los datos del Data Warehouse Empresarial, y los prepara para entregar a la data marts independientes, por ejemplo: una data mart de ventas, de compras, de contabilidad, etc…

Finalmente, las aplicaciones de BI consumen estos datos ya sea de la data mart o del Data Warehouse Empresarial.

- **Arquitectura de Datamarts independientes**

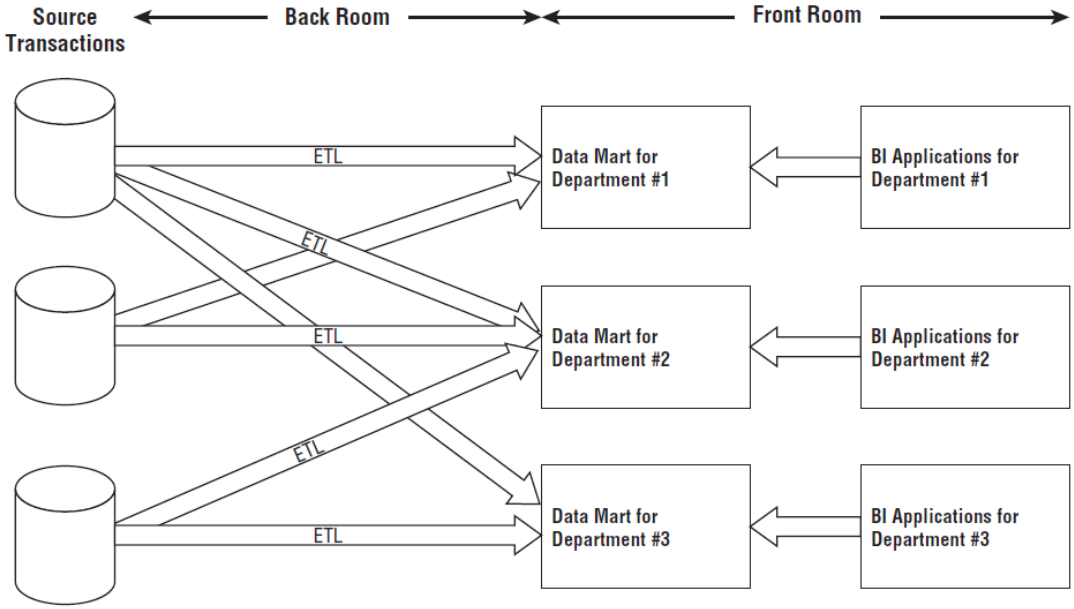

Figura 10: Arquitectura de DataMarts independientes

Esta arquitectura es usada por ciertas compañías, para poner sus datos a disposición, hacer componentes de extracción de datos que llevan estos datos a repositorios comunes pero divididos por unidad de negocio y ahí crean la cantidad de datos requeridos para que una aplicación de BI puntual para un proceso de negocio la consuma.

Cuadro comparativo de ventajas y desventajas de arquitecturas de data warehouse

Arquitectura	Ventajas	Desventajas
Arquitectura de Kimball	• El modelado dimensional de Kimball es rápido de construir ya que no implica normalización, lo que significa una rápida ejecución de la fase inicial del almacenamiento de datos de procesos.	• Pueden ocurrir irregularidades cuando los datos se actualizan en la arquitectura Kimball DW. Esto se debe a que, en la técnica de des normalización, se agregan datos redundantes a las tablas de la base de datos.

Arquitectura	Ventajas	Desventajas
	• Una ventaja del esquema en estrella es que la mayoría de los operadores de datos pueden comprenderlo fácilmente debido a su estructura des normalizada, que simplifica las consultas y el análisis. • Un equipo más pequeño de diseñadores y planificadores es suficiente para la gestión del almacén de datos porque los sistemas de origen de datos son estables y el almacén de datos está orientado a procesos. Además, la optimización de consultas es sencilla, predecible y controlable. • Permite la recuperación rápida de datos del almacén de datos, ya que los datos se segregan en tablas de hechos y dimensiones. • Es rápido de implementar	• En la arquitectura Kimball DW, pueden ocurrir problemas de rendimiento debido a la adición de columnas en la tabla de hechos, ya que estas tablas son bastante detalladas. La adición de nuevas columnas puede expandir las dimensiones de la tabla de hechos, lo que afecta su rendimiento. Además, el modelo de almacén de datos dimensional se vuelve difícil de modificar con cualquier cambio en las necesidades comerciales. • Como el modelo de Kimball está orientado a los procesos comerciales, en lugar de centrarse en la empresa en su conjunto, no puede manejar todos los requisitos de informes de BI.

Arquitectura	Ventajas	Desventajas
Arquitectura de Inmon	• El almacén de datos actúa como una fuente de verdad unificada para todo el negocio, donde todos los datos están integrados. • Este enfoque tiene una redundancia de datos muy baja. Por lo tanto, hay menos posibilidades de irregularidades en la actualización de datos, lo que hace que el proceso de almacenamiento de datos basado en el concepto ETL sea más sencillo y menos susceptible a fallas. • Este enfoque ofrece una mayor flexibilidad, ya que es más fácil actualizar el almacén de datos en caso de que haya algún cambio en los requisitos comerciales o en los datos de origen. • Puede manejar diversos requisitos de informes en toda la empresa.	• La complejidad aumenta a medida que se agregan varias tablas al modelo de datos con el tiempo. • Se requieren recursos capacitados en el modelado de datos de almacenamiento de datos, que pueden ser costosos y difíciles de encontrar. • La configuración preliminar y la entrega requieren mucho tiempo. • Se requiere una operación de proceso ETL adicional ya que los data marts se crean después de la creación del almacén de datos. • Este enfoque requiere que los expertos administren un almacén de datos de manera efectiva. • Demanda más tiempo de implementación.

Arquitectura	Ventajas	Desventajas
Arquitectura de data marts independientes	• Al reducir el volumen de datos, una despensa de datos ayuda a mejorar el tiempo de respuesta del usuario y ofrece un acceso rápido a los datos de uso frecuente. • Es fácil de implementar con un costo mucho menor, en comparación con la implementación de un almacén de datos completo. • Es escalable y ágil, lo que resulta útil a la hora de cambiar de modelo. • Los datos se pueden almacenar y organizar en distintas plataformas de hardware o software	• Repetición de procesos ETL, debido a que se puede requerir datos que ya extrae un ETL, pero el destino de los datos es un repositorio o data mart diferente, por lo que se tiene que duplicar el ETL para ese caso de uso específico. • Por consecuencia de lo anterior, el mantenimiento de este tipo de arquitecturas se vuelve insostenible con el tiempo. • Difíciles de administrar. Esto se debe a que los analistas de negocios deben realizar tareas administrativas de bases de datos en cada datamart. • Requieren un trabajo de configuración especializado, ya que cuentan con un sistema exclusivo de almacenaje. • Por lo anterior, esto también puede tener

Arquitectura	Ventajas	Desventajas
		como consecuencia un mayor gasto. • Por otro lado, al migrar la información a una base de datos general, se debe adecuar su formato y, si hablamos de volúmenes de información significativos, puede ser un proceso tardado.

Con base en la comparativa presentada, se descartó la opción de Data marts independientes debido a que la solución que se propone está pensada para ser utilizada por mucho tiempo y esta opción se vuelve muy complicada en ese caso. Además, esta opción de Data Marts independientes es difícil de administrar, y la propuesta de solución busca facilitar en lugar de dificultar las tareas a los usuarios. Por estas razones, se descartó esta opción.

Entre la arquitectura de Bill Inmon y la de Ralph Kimball, se analizaron las ventajas y desventajas de cada una, además de su aplicabilidad en la propuesta de solución. Cuando se trata de informar necesidades, Kimball permita la creación de informes centrados en el proceso comercial; en cambio Inmon permite crear informes integrados de toda la organización. Aplicada esta ventaja a la propuesta de solución, la arquitectura de Kimball es la mejor opción debido a que la propuesta se centra en el área de ventas, y no en todas las áreas de la empresa.

Otra de las ventajas sobre el método Kimball sobre Inmon es que el enfoque de Inmon propone un modelo de datos normalizado, lo cual es más complejo de diseñar que

un modelo des normalizado, que es el tipo de modelo propuesto por Kimball. Esto hace que el enfoque de Inmon sea un proceso que requiere mucho tiempo. En cambio, el enfoque de Kimball es más aplicable a la propuesta de solución que se busca realizar en menos tiempo.

Por otro lado, la mayor complejidad de la creación de modelos de datos utilizando el modelo de Inmon requiere un equipo más grande de profesionales para la gestión del almacén de datos. Debido a que la propuesta de solución busca reducir costos, que en este caso sería la necesidad de un equipo más grande, el enfoque de Kimball es la mejor opción.

Por todas estas razones, se escogió el enfoque de Ralph Kimball como la arquitectura de data warehouse a emplear en la solución propuesta.

Análisis de los sistemas fuentes y de las métricas

Modelado dimensional

En esta sección, se ejecutará el proceso de análisis dimensional para poder definir a alto nivel el esquema a construir en la etapa de presentación.

Primero, se selecciona el proceso de negocio. En este caso, el proceso de negocio seleccionado es el proceso de ventas omnicanal.

El siguiente paso es definir la granularidad o grado de detalla. Dado que muchos de los requerimientos implican saber qué tanta ganancia se obtuvo para un producto o qué tanto se vendió. Se ha optado por definir el nivel de detalle por producto, tomando en cuenta que cada registro de venta podrá dársele seguimiento por día como mínimo.

Habiendo definido la granularidad, se procede a identificar las dimensiones. Las dimensiones identificadas son: Dimensión de productos, dimensión de dirección de clientes, dimensión de tiempo, dimensión de tienda, y dimensión de clientes.

Con las dimensiones ya identificadas, se procede a identificar las métricas.

Definición de las métricas

- A la empresa le gustaría conocer la cantidad de producto vendido por tienda
- A la empresa le gustaría determinar el monto vendido por tienda
- A la empresa le gustaría determinar el top 5 de clientes que más han comprado en el negocio
- A la empresa le gustaría saber el top 5 de productos que más se han vendido en el negocio
- A la empresa le gustaría saber cuál es el monto de venta generado por marca de producto
- A continuación, se presenta el diagrama de estrella a alto nivel:

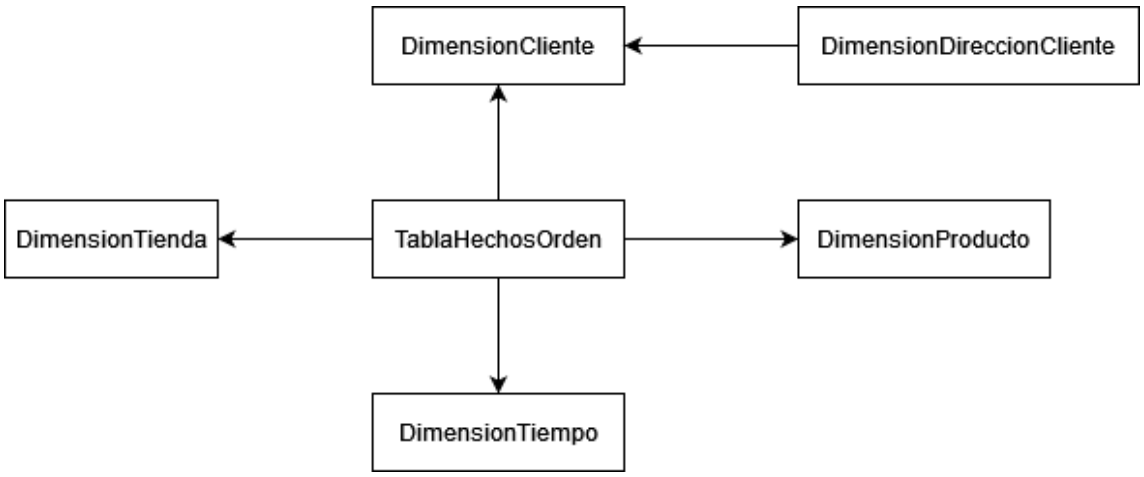

Figura 11: Diagrama de estrella a alto nivel

Análisis de los sistemas fuentes

En esta sección, se analizará los esquemas y tablas del sistema origen que se necesitan para realizar la extracción, limpieza, procesamiento y carga de datos en las tablas respectivas del modelo dimensional.

Para la dimensión de producto, se necesitan los siguientes campos provenientes del sistema origen.

Nombre de la tabla	Nombre del campo	Tipo de dato	Nulo	Valor predeterminado	Renombrado como
product	price	Decimal(20, 6)	No	0.00	
product_lang	name	Varchar(128)	No	Ninguno	
category_lang	name	Varchar(128)	No	Ninguno	category
product_attibute	Id_product	Int(10)	No	Ninguno	
manufacturer	name	Varchar(64)	No	Ninguno	brandname
attribute_group_lang	Name	Varchar(128)	No	Ninguna	attributesname
attribute_lang	name	Varchar(64)	No	Ninguna	attributevaluename
feature_lang	Name	Varchar(64)	No	Ninguna	featurename
feature_value_lang	value	Varchar(255)	Sí	Nulo	featurevaluename

Para la dimensión de cliente, se necesitan los siguientes campos provenientes del sistema origen.

Nombre de la tabla	Nombre del campo	Tipo de dato	Nulo	Valor predeterminado	Renombrado como
customer	id_customer	Int(10)	No	Ninguno	
customer	firstname	Varchar(255)	No	Ninguno	
customer	lastname	Varchar(255)	No	Ninguno	
customer	id_gender	Int(10)	No	Ninguno	
customer	birthday	Date	Sí	Nulo	
customer	email	Varchar(255)	No	Ninguno	
customer	is_guest	Tinyint(1)	No	0	
customer	date_add	Datetime	No	Ninguno	

Para la dimensión de dirección de cliente, se necesitan los siguientes campos provenientes del sistema origen.

Nombre de la tabla	Nombre del campo	Tipo de dato	Nulo	Valor predeterminado	Renombrado como
address	id_address	Int(10)	No	Ninguno	
address	id_customer	Int(10)	No	0	
address	company	Varchar(255)	Sí	Nulo	
address	address1	Varchar(128)	No	Ninguno	
address	address2	Varchar(128)	Sí	Nulo	

address	postcode	Varchar(12)	Sí	Nulo	
address	city	Varchar(64)	No	Ninguno	
address	phone	Varchar(32)	Sí	Nulo	
address	phone_mobile	Varchar(32)	Sí	Nulo	
address	deleted	Tinyint(1)	No	1	
country_lang	name	Varchar(64)	No	Ninguno	country
state	name	Varchar(80)	No	Ninguno	state

Para la dimensión tienda, se necesitan los siguientes campos provenientes del sistema origen.

Nombre de la tabla	**Nombre del campo**	**Tipo de dato**	**Nulo**	**Valor predeterminado**	**Renombrado como**
shop	id_shop	Int(11)	No	Ninguno	
shop	name	Varchar(64)	No	Ninguno	

Para la tabla de hechos de orden, se necesitan los siguientes campos provenientes del sistema origen.

Nombre de la tabla	**Nombre del campo**	**Tipo de dato**	**Nulo**	**Valor predeterminado**	**Renombrado como**
orders	id_order	Int(10)	No	Ninguno	
orders	id_customer	Int(10)	No	Ninguno	
orders	id_shop	Int(11)	No	1	
orders	payment	Varchar(255)	No	Ninguno	
orders	total_shipping_tax_incl	Decimal(20, 6)	No	0.00	
orders	total_discounts	Decimal(20, 6)	No	0.00	
orders	date_add	Datetime	No	Ninguno	
orders	date_upd	Datetime	No	Ninguno	
order_state_lang	name	Varchar(64)	No	Ninguno	current_state
order_detail	product_id	Int(10)	No	Ninguno	

order_detail	product_quantity	Int(10)	No	0	
order_detail	unit_price_tax_incl	Decimal(20, 6)	No	0.00	

De las tablas anteriores, ya es posible saber cuál es la estructura de los datos e identificar cuál es la calidad de los mismos, en el apartado de procesamiento de datos se tomará en cuanta esto para realizar la limpieza de datos y procesamiento, pero ya es posible concluir que las métricas requeridas son posibles de contestar con los datos disponibles.

Modelo dimensional detallado

Ahora que sabemos la estructura de cada campo a usar en el modelo dimensional nuestro diagrama de alto nivel mostrado en la figura 12, quedaría detallado de la siguiente manera:

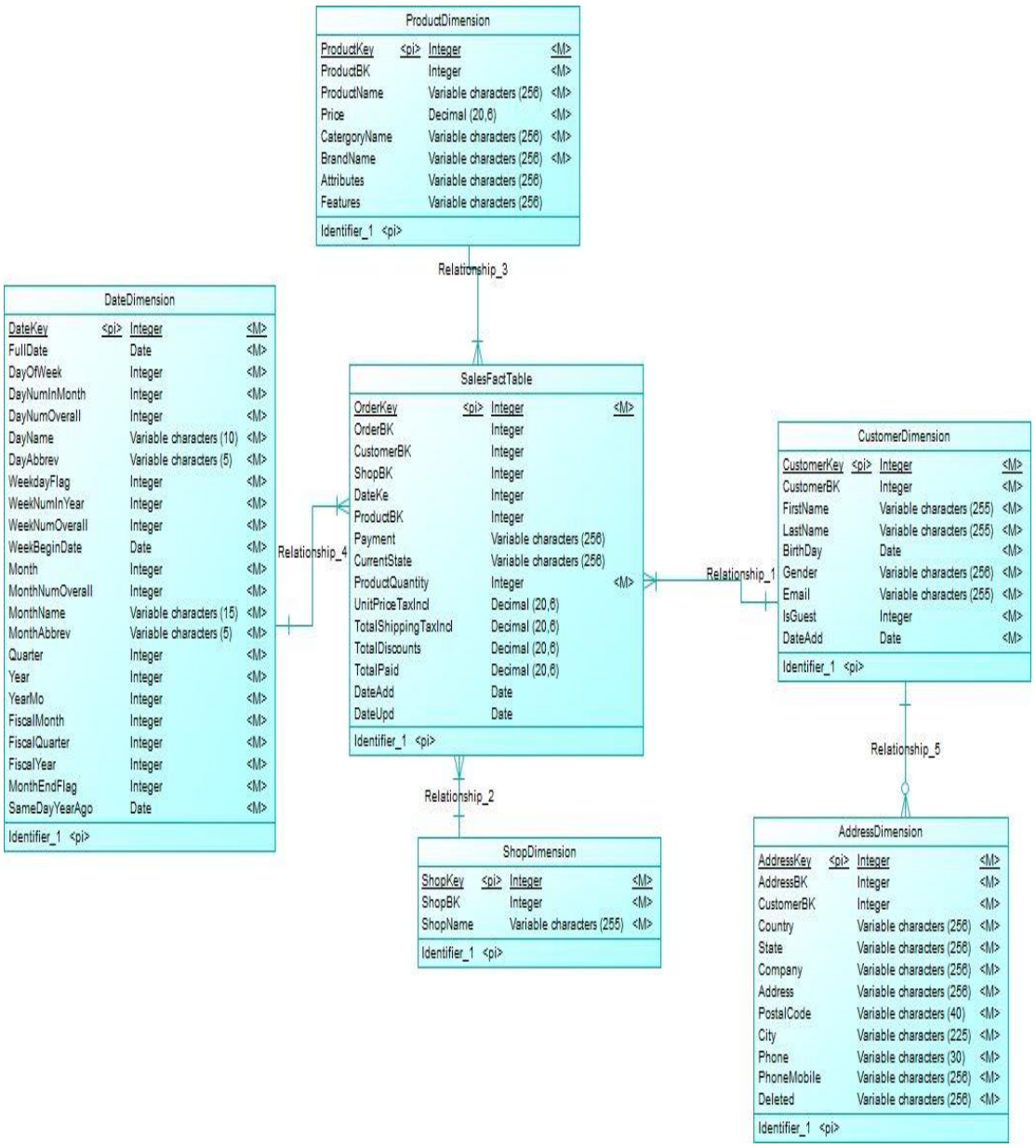

Figura 12: Modelo dimensional detallado

Procesamiento de datos (Extracción, Transformación y Carga de datos)

A continuación, describimos como se realizó el procesamiento de datos el cual comprende las etapas de extracción, transformación y carga de datos:

Configuración previa

Para llevar a cabo la correcta extracción, transformación y carga de datos, se realizó una configuración previa en el sistema origen, que consiste de una base de datos con dos tablas que tiene la siguiente estructura:

Nombre de base de datos: distribuidora_el_aguila_configs

Nombre de la tabla: jobs_config			
Nombre campo	Tipo de dato	Nulo	Predeterminado
id	bigint(20)	No	Ninguna
key	varchar(150)	No	Ninguna
value	varchar(255)	No	Ninguna
date_add	date	Sí	current_timestamp()
date_upd	date	Sí	current_timestamp()

Esta tabla es empleada para definir variables usadas en el contexto de cada proceso ETL ejecutado, esta tabla deberá contener ciertos registros antes de la ejecución de los procesos ETL, para que puedan ejecutarse con normalidad, estos valores se definen más adelante junto con el proceso ETL que carga estos valores.

Nombre de la tabla: jobs_exec_logs			
Nombre campo	Tipo de dato	Nulo	Predeterminado
moment	datetime	No	*Ninguna*
pid	varchar(20)	No	*Ninguna*
root_pid	varchar(20)	No	*Ninguna*
father_pid	varchar(20)	No	*Ninguna*
project	varchar(50)	No	*Ninguna*
job	varchar(255)	No	*Ninguna*
context	varchar(50)	No	*Ninguna*
priority	int(3)	No	*Ninguna*
type	varchar(255)	No	*Ninguna*
origin	varchar(255)	No	*Ninguna*

message	varchar(255)	No	*Ninguna*
code	int(3)	No	*Ninguna*

Esta tabla se usa para registrar cualquier falló que sea detectado al momento de ejecutar un proceso ETL, el error es manejado y registrado en una tabla con la estructura de la tabla de arriba.

Extracción de Datos

En este apartado se describe de manera resumida cómo se extraen los datos que solicita el modelo dimensional para su posterior transformación.

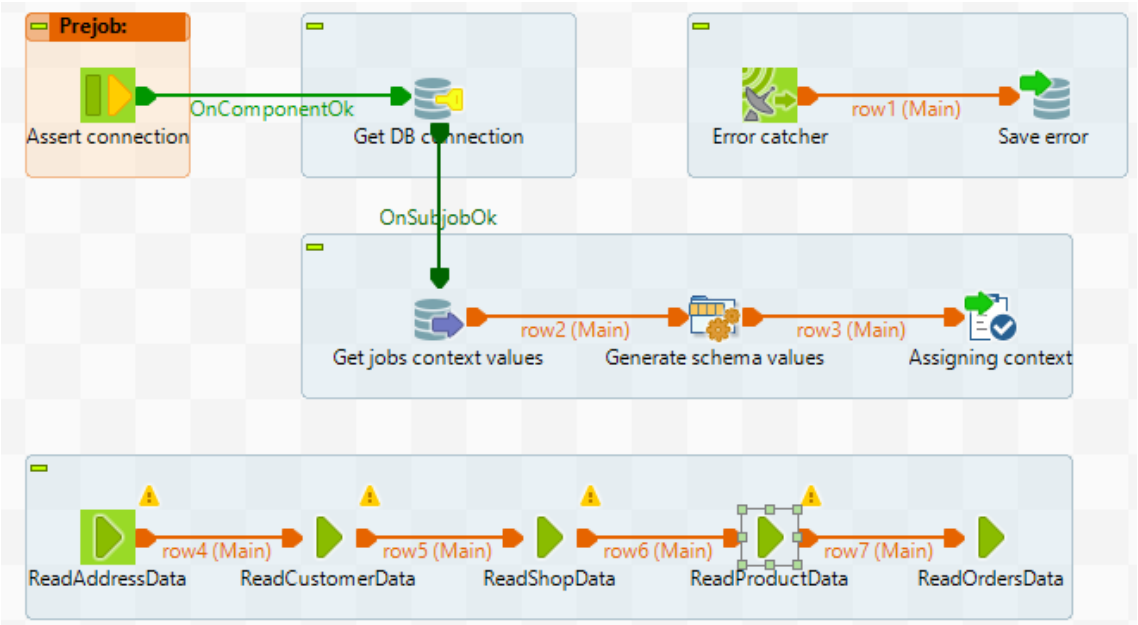

Figura 13: Proceso etl maestro de extracción de datos

En la imagen de arriba puede observarse el proceso ETL maestro que se encarga de la extracción de datos de todas las tablas del sistema origen que se necesitan para el su posterior transformación y carga en el modelo dimensional creado.

Este proceso ETL, inicialmente configura los valores necesarios para la ejecución de los procesos de extracción de datos. Los valores que se asignan al contexto de cada proceso ETL son los que se observan a continuación:

Valores de configuración de procesos etl		
Nombre de la propiedad	Valor	Descripción
CustomerDimension	38	Último valor generado para la llave subrogada de la dimensión de customer.
AddressDimension	37	Último valor generado para la llave subrogada de la dimensión de address.
ProductDimension	40	Último valor generado para la llave subrogada de la dimensión de product.
ShopDimension	2	Último valor generado para la llave subrogada de la dimensión de shop.
ServerFilePath	C:/Users/Escritorio/UES/Ingenieria/	Ruta base para almacenamiento de CSV generados.
CustomerDimFileName	CustomerDimension.csv	Nombre del archivo CSV a asignar para los datos de la dimensión de customer
AddressDimFileName	AddressDimension.csv	Nombre del archivo CSV a asignar para los datos de la dimensión de address
ProductDimFileName	ProductDimension.csv	Nombre del archivo CSV a asignar para los datos de la dimensión de product
ShopDimFileName	ShopDimension.csv	Nombre del archivo CSV a asignar para los datos de la dimensión de shop
OrdersFactFileName	OrdersFact.csv	Nombre del archivo CSV a asignar para los datos de la dimensión de orders
OrdersDimension	17	Último valor generado para la llave subrogada de la dimensión de orders.
LastJobExecDateOrders	7/1/2023 03:24	Última fecha de ejecución para el proceso de orders.
LastJobExecDateAddress	7/1/2023 03:24	Última fecha de ejecución para el proceso de address.

LastJobExecDateCustomer	7/1/2023 03:24	Última fecha de ejecución para el proceso de customer.
LastJobExecDateShop	7/1/2023 03:24	Última fecha de ejecución para el proceso de shop.
LastJobExecDateProduct	7/1/2023 03:24	Última fecha de ejecución para el proceso de product.

Valores de configuración de componentes AWS		
Nombre de la propiedad	Valor	Descripción
RawFolderName	raw/	Nombre de la carpeta donde se guarda los datos crudos.
StagingFolderName	staging/	Nombre de la carpeta donde se guarda los datos transformados.
PresentationFolderName	presentation/	Nombre de la carpeta en S3 donde se cargan los archivos CSV generados.
AWSAccessKeyId	AKIARRBPW2TGJWOZH75L	Access key al bucket S3.
AWSSecretKey	HzERboFSIQDDyADOGjqcm	Secret key al bucket S3.
BucketName	files-el-aguila	Nombre del Bucket.
S3FolderName	presentation/	Nombre de la carpeta en S3 donde se cargan los archivos CSV generados.
NewDataFolder	new-data/	Nombre de la carpeta en S3 donde se guarda los CSV con datos nuevos.
UpdateDataFolder	updated-data/	Nombre de la carpeta en S3 donde se guarda los CSV con datos a actualizar.

El estado de los datos expuesto es un ejemplo de la configuración de estos campos, estos valores se asignan como variables de contexto en el componente que aparece en la imagen nombrado como «Assigning context».

Luego de la lectura de los datos de contexto, se ejecuta uno a uno secuencialmente los procesos de extracción de datos del sistema origen. Para resumir el cómo funcionan estos procesos se explica a continuación uno de ellos dado que todos realizan un proceso similar con la única diferencia siendo los datos que extraen para inserción o actualización en el modelo dimensional. Cabe mencionar que no existe un proceso ETL para la extracción, transformación y carga de datos de la dimensión de tiempo nombrada

«datedimension» ya que se trata de una dimensión precargada, de modo que se agrega o actualiza datos directamente en el esquema estrella.

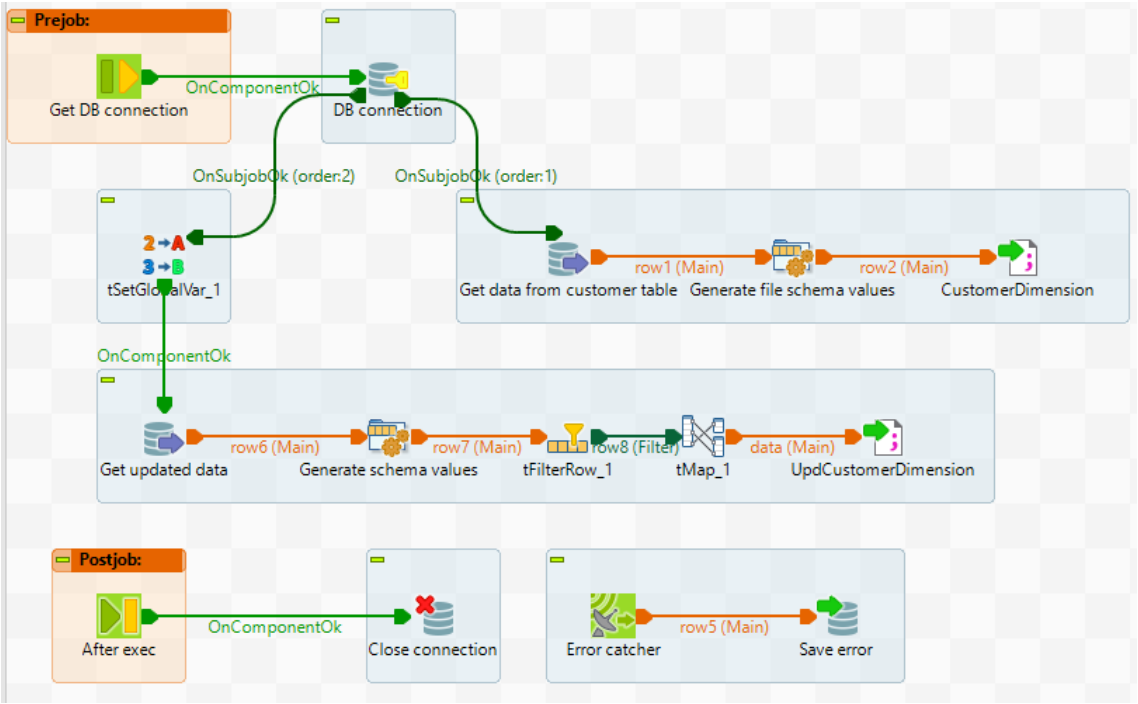

Figura 14: Proceso etl para extracción de datos de clientes

Inicialmente el proceso ETL asegura la conexión con el sistema origen, luego de asegurar la conexión ejecuta dos flujos:

- El primer flujo se encarga de leer solo los datos nuevos que han sido ingresados en la tabla o tablas del sistema origen, esto usando la fecha en que fue agregado el registro comparado contra la última fecha en que se ejecutó el proceso ETL, de esta manera distingue que datos ya han sido leídos anteriormente, luego de esto genera un esquema para con los datos para poder ser almacenado en formato CSV en el servidor local.
- El segundo flujo se encarga de leer solo datos que necesitan ser actualizados usando un criterio similar al del flujo anterior con la diferencia que esta vez se usa la fecha de actualización del registro, inicialmente solo lee los datos sin usar en la consulta SQL algún filtro en la cláusula WHERE, luego de la lectura se genera el esquema para almacenar en el archivo CSV, es en el siguiente paso que se filtra los datos por la fecha de actualización del registro por lo que únicamente pasan al archivo CSV aquellos registros cuya fecha de

actualización es mayor a la última fecha de ejecución del proceso ETL, finalmente se almacenan esos registros en un archivo CSV.

Al finalizar los flujos de extracción de datos se cierra la conexión al sistema origen, y si surge algún error o se lanza alguna excepción durante la ejecución este es manejado por el componente nombrado «Error catcher» y se registra en la tabla de errores de ejecución descrita en el apartado de configuración previa.

Lo anterior descrito aplica para todos los procesos de extracción de datos.

Transformación de datos

En este apartado se describe como se procesa los datos para su posterior carga, se describe el proceso ETL maestro que invoca todos los procesos de transformación y se describe un proceso de transformación ya que es lo mismo para todos con la diferencia en los tipos de datos que se transforma.

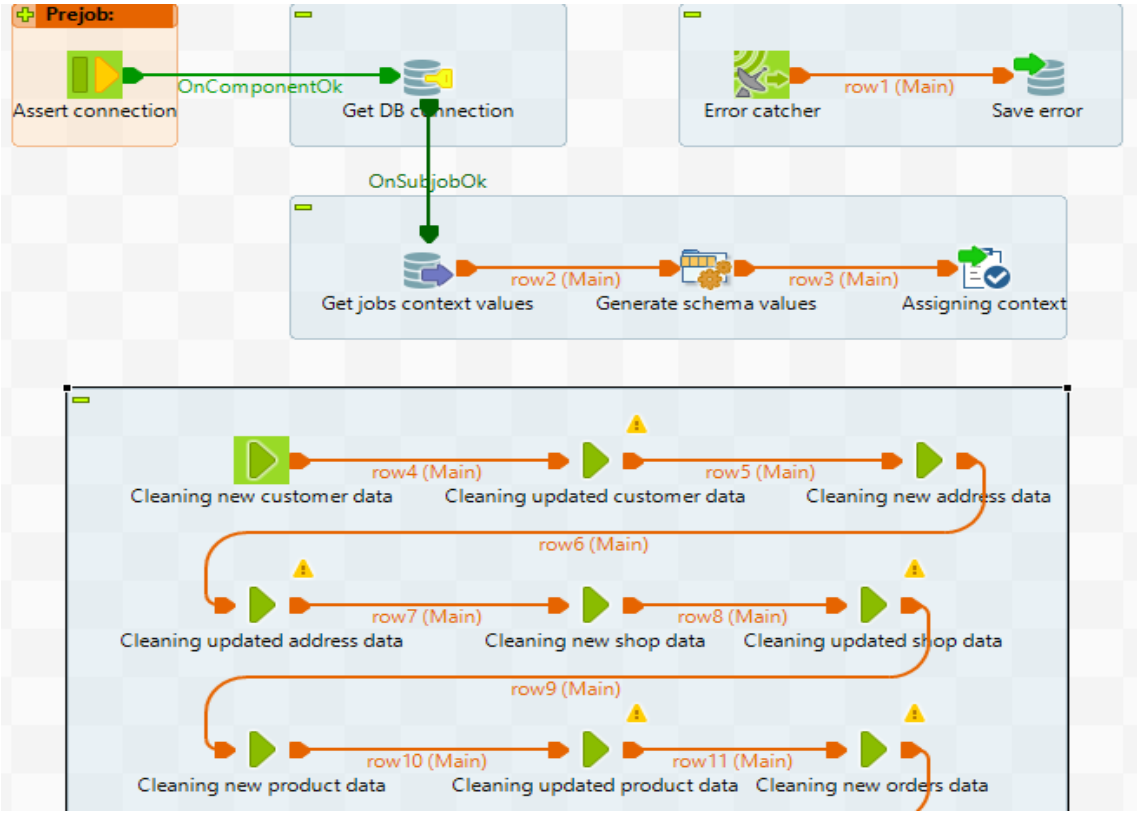

Figura 15: Proceso etl maestro para transformación de datos

Al igual que el proceso maestro de extracción de datos, el proceso maestro de transformación también carga los valores necesarios para la ejecución de los procesos ETL, luego invoca uno a uno de manera secuencial los procesos de transformación tanto

para los archivos CSV que contienen datos nuevos como para los archivos CSV que contienen datos a actualizar.

A continuación, se describe un proceso ETL para transformación que sirve para explicar el flujo de todos los que se invocan en el proceso maestro.

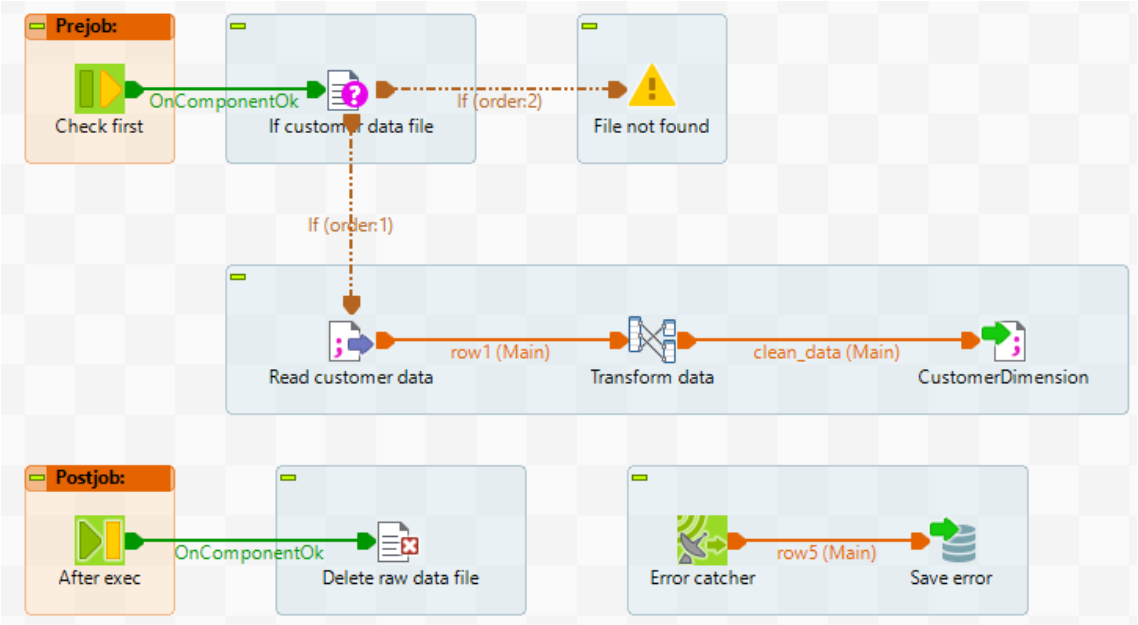

Figura 16: Proceso etl de transformación de datos de clientes

El proceso de transformación de los datos sigue el flujo mostrado en la Figura 16, inicialmente el proceso se asegura que el archivo CSV que debe procesar existe, si este no existe se lanza una excepción no bloqueante para ser registrada en la tabla de ejecución de los procesos.

Pero si el archivo existe entonces se lee el archivo, luego con el componente tMap nombrado como «Transform data» se realizan conversiones de datos, verificación y transformación de datos nulos a valores predefinidos que cumplan el esquema destino, y esta data limpia y transformada se almacena en un nuevo archivo CSV.

Finalmente, se elimina el archivo de datos crudos del servidor, dado que ya no se necesita usar. Al igual que el proceso de extracción, el de transformación también cuenta con los componentes para registro de errores.

Carga de datos

En este apartado se describe los procesos ETL usados para la carga de datos, se abordará más sobre los componentes de nube AWS usados en este apartado cuando se describa el área de presentación.

A continuación, se describe el proceso maestro de carga de datos.

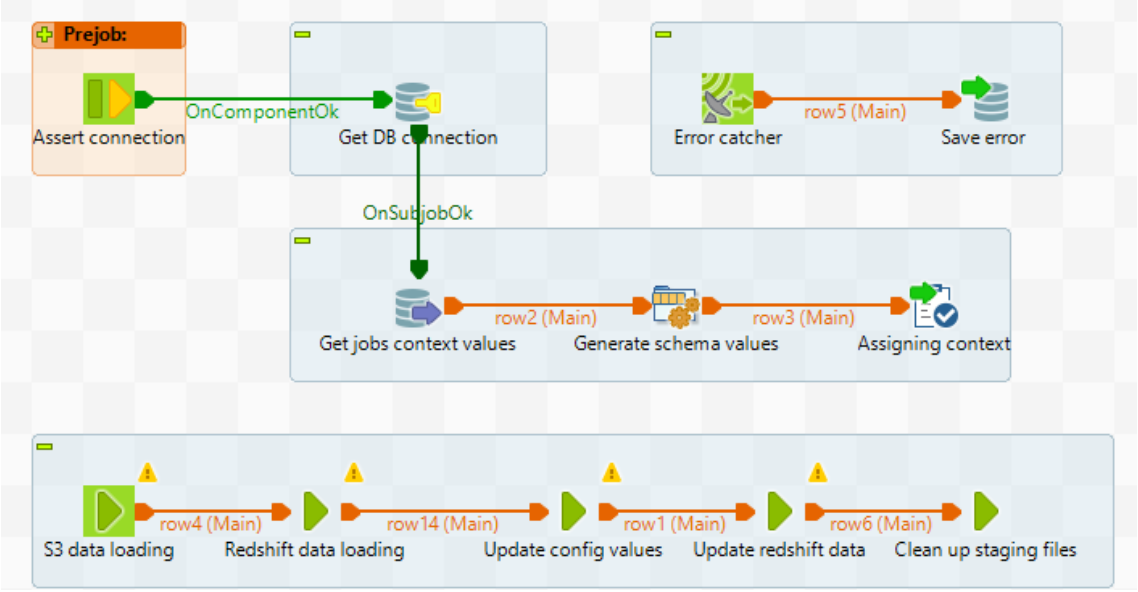

Figura 17: Proceso ETL maestro para carga de datos

En la imagen de la Figura 17, se muestra el proceso ETL maestro de carga de datos, al igual que los anteriores esta carga en el contexto de cada proceso los datos necesarios para su ejecución, y posteriormente se ejecuta secuencialmente los procesos ETL para cargar los datos en los componentes de Amazon elegidos. Se describe a continuación los procesos ETL definidos para la carga de datos en Amazon S3, Amazon Redshift y para la actualizar los datos de configuración para la siguiente ocasión en que se ejecute el proceso ETL.

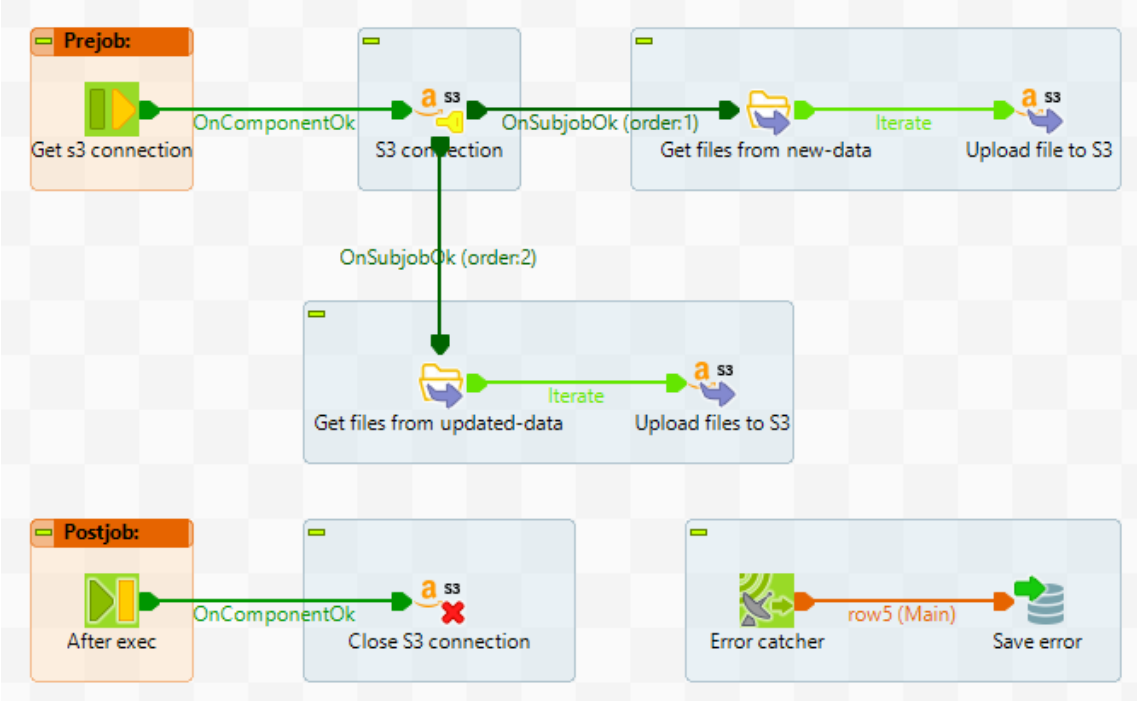

Figura 18: Proceso etl para carga de archivos en S3

En el proceso ETL que se muestra en la imagen de la Figura 18, inicialmente se asegura la conexión con el bucket de S3 al cual se cargarán los archivos de datos, luego de tener una conexión exitosa, se ejecutan dos flujos:

- El primer flujo se encarga de buscar y cargar en S3, todos los archivos CSV que contienen nuevos datos extraídos del sistema origen.
- El segundo flujo se encarga de buscar y cargar en S3, todos los archivos CSV que contienen datos actualizados del sistema origen.

Finalmente, se cierra la conexión al bucket de S3.

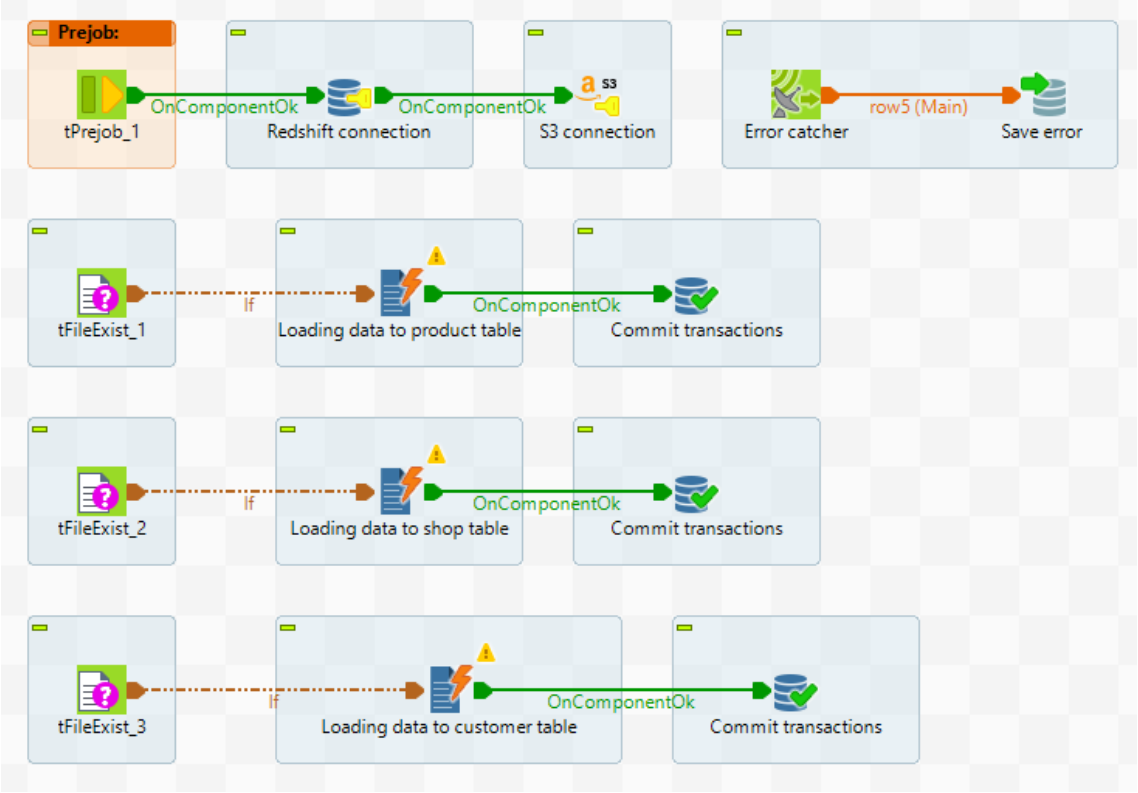

Figura 19: Proceso etl para carga de datos en Amazon Redshift

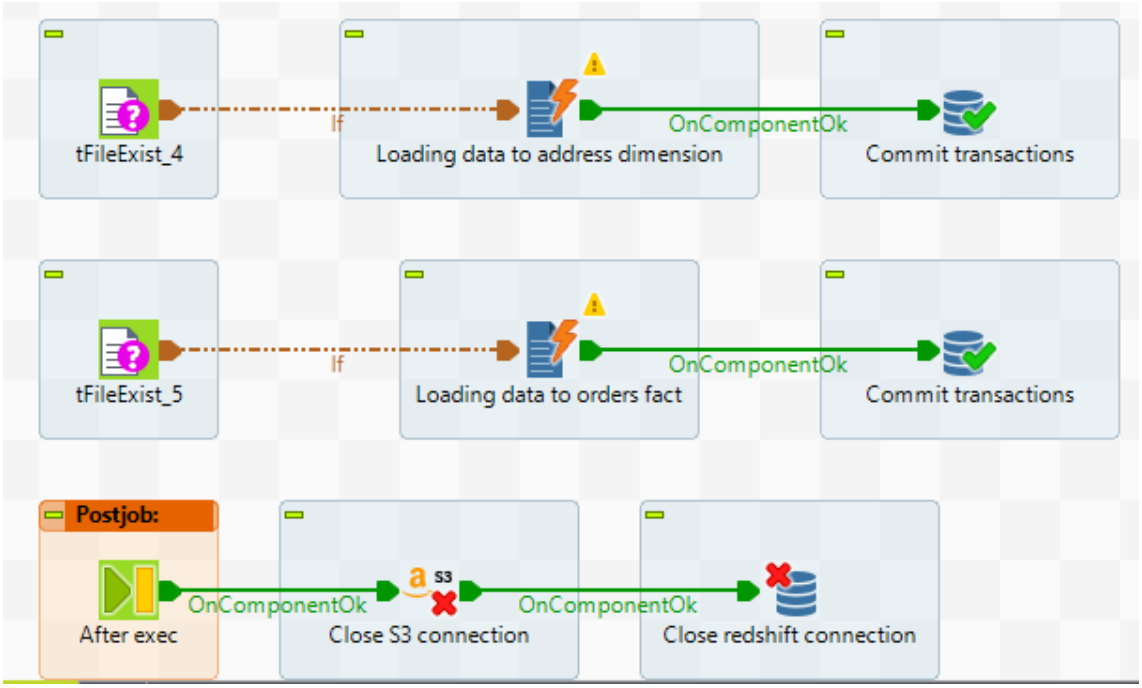

Figura 20: Proceso etl para carga de datos en Amazon Redshift

En la Figura 19 y 20, se muestra el proceso ETL que carga los datos desde Amazon S3 a Amazon Redshift.

Primero se asegura la conexión con el bucket de S3 y con Amazon Redshift, posteriormente se verifica si existe para cada dimensión y tabla de hechos un archivo que necesite ser cargado en el esquema creado en Amazon Redshift, si existe un nuevo archivo a cargar entonces se realiza la carga de este desde S3, y se ejecuta un componente un commit para finalizar la transacción.

Esto se hace para cada dimensión y tabla de hechos, al finalizar se cierra la conexión con S3 y Amazon Redshift.

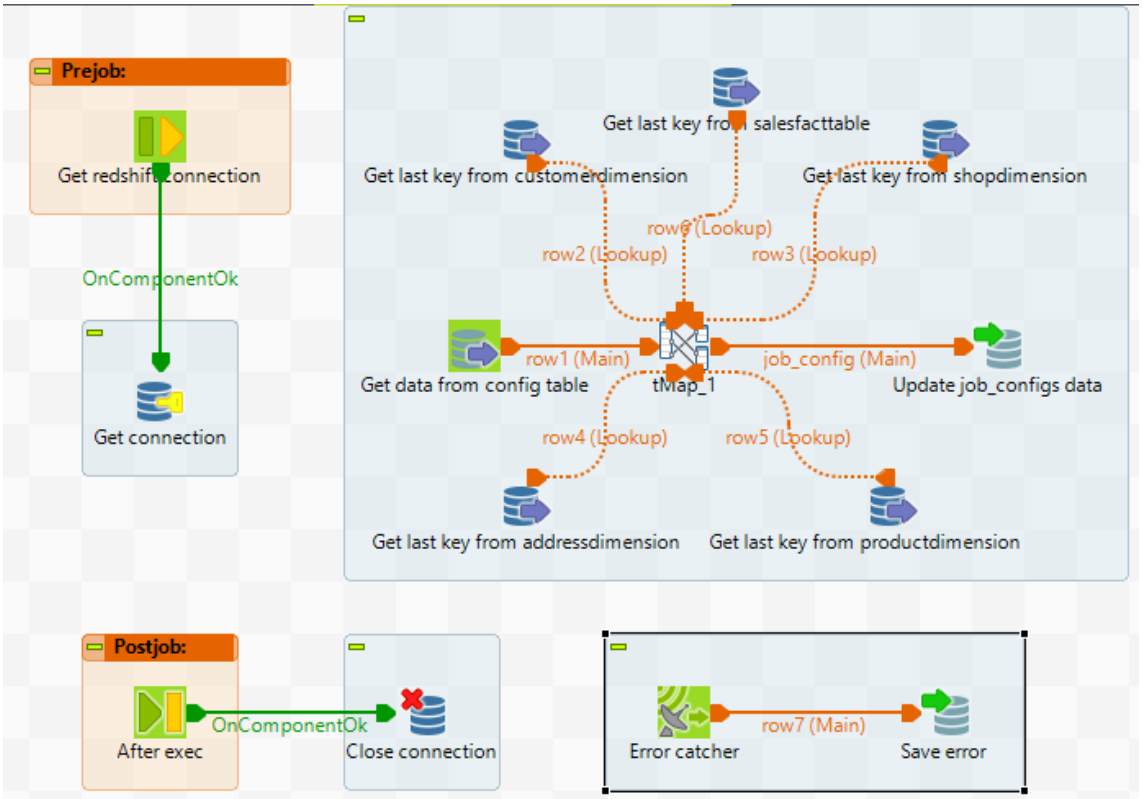

Figura 21: Proceso etl para actualización de valores de configuración

El proceso ETL mostrado en la Figura 21, se encarga de actualizar los valores de configuración de almacenados en el sistema origen, en este caso se trata de obtener y almacenar el último valor generado para la llave subrogada de cada dimensión del esquema de estrella creado.

Primero se obtiene una conexión con Amazon Redshift, luego se lee los datos de la tabla de configuración creada en la base de datos para valores de configuración, y de cada dimensión, a excepción de la dimensión de tiempo, se obtiene el último valor para el campo de llave subrogada de la tabla.

Mediante el uso de un tMap, se identifica que valores hay que actualizar y una vez hecho se guardan los nuevos valores en la tabla de configuración, esto con el objetivo que la siguiente ejecución genere nuevos valores para las llaves subrogadas y no valores repetidos.

Finalmente se cierra la conexión con Amazon Redshift.

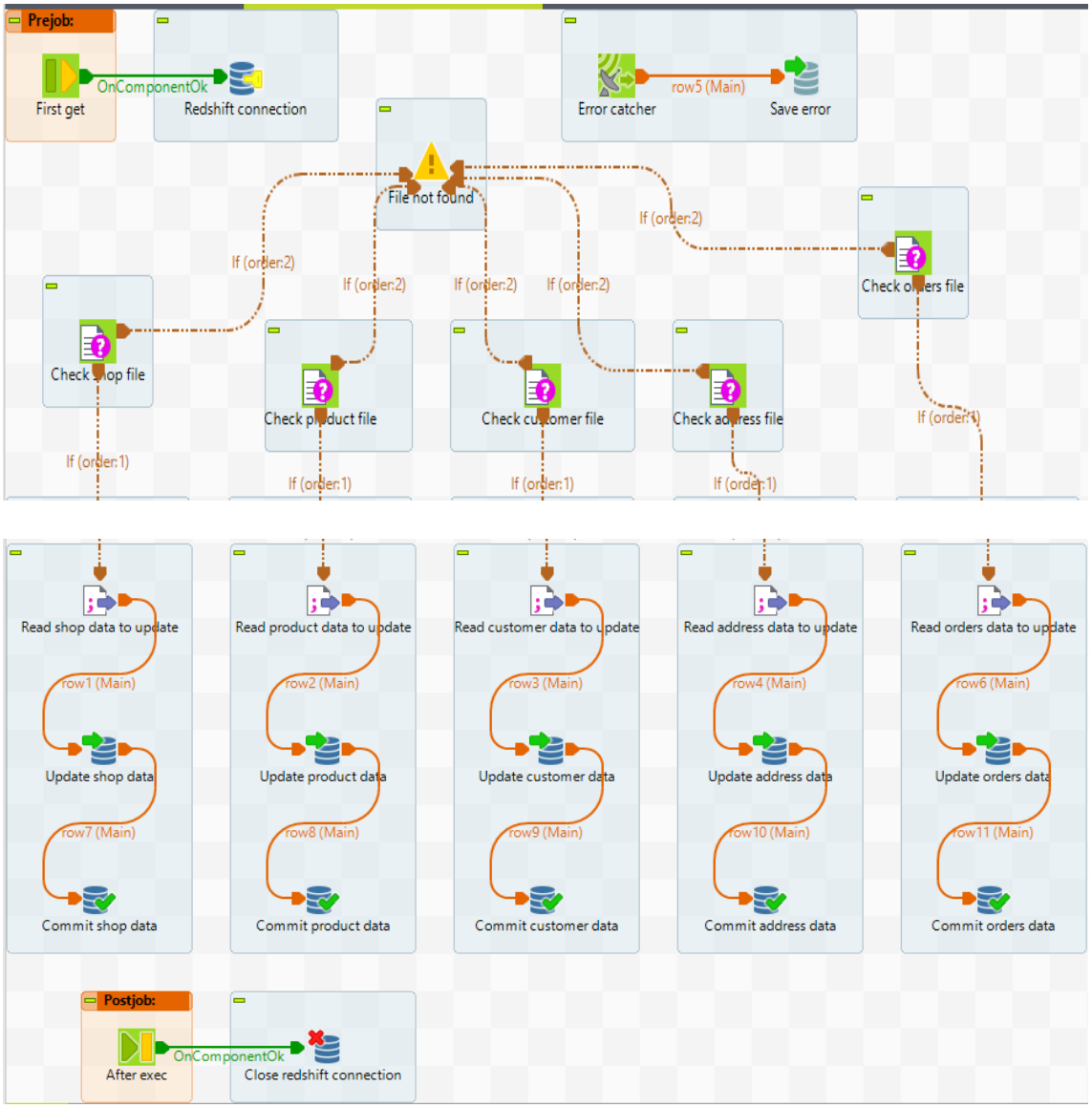

Figura 22: Proceso etl para actualizar datos en Amazon Redshift

El proceso etl mostrado en la Figura 22, se encarga de hacer actualización de los datos en el esquema estrella creado, cuando se han generado actualizaciones en los datos de la base de datos origen.

El proceso inicia conectándose a Amazon Redshift, luego verifica si existen archivos nuevos con datos para actualizar, esto por cada dimensión y tabla de hechos

excepto para la dimensión de tiempo, si existe un archivo entonces se realiza la actualización mediante un componente de salida a Amazon Redshift con la operación definida para actualizar datos.

Finalmente, luego de todas las actualizaciones, se cierra la conexión con Amazon Redshift.

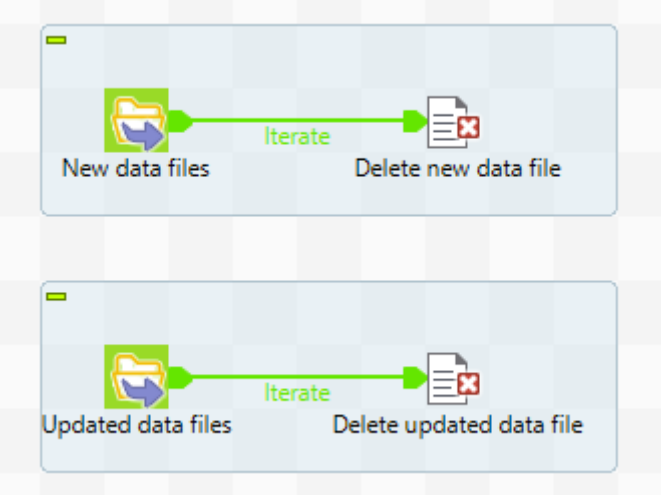

Figura 23: Proceso etl para limpiar los archivos CSV generados

El proceso etl mostrado en la Figura 23, solo tiene como tarea eliminar los archivos CSV generados al finalizar todos los etl anteriormente descritos.

Área de presentación de datos

En esta sección se definen los componentes en la nube de Amazon AWS utilizados y el motivo por el cual son utilizados para la presentación de los datos.

Amazon S3

El servicio Amazon S3 se utiliza como un lago de datos para almacenar todos los datos provenientes del sistema origen. Además, como posible respaldo de dichos datos. De igual forma, se utiliza como intermediario para cargar de datos el esquema de estrella utilizado en Amazon Redshift.

Se creó un bucket, que es un contenedor de objetos, para poder almacenar todos los datos. En dicho bucket, se configuró una estructura de carpetas como puede notarse en las siguientes imágenes.

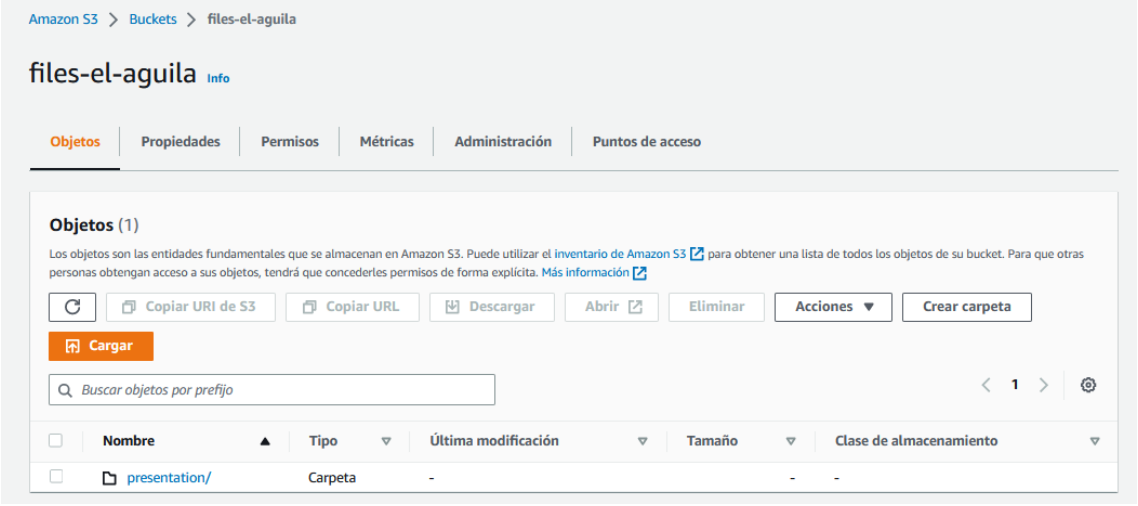

Figura 24: Bucket en Amazon S3

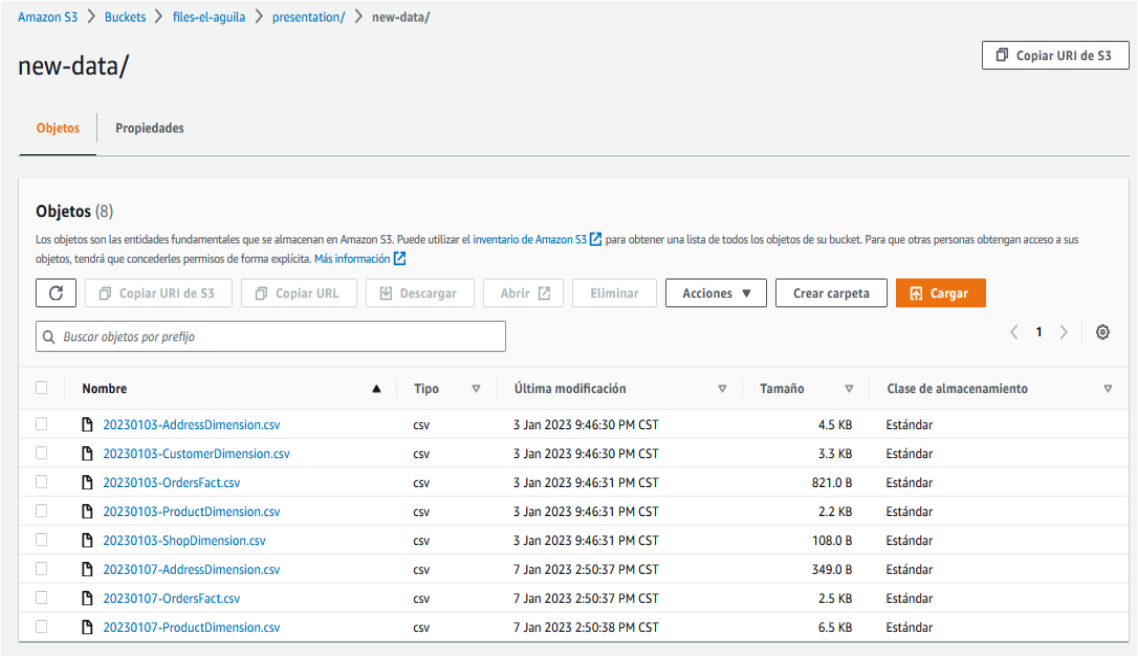

Figura 25: Archivos cargados en Amazon S3

Amazon Redshift

El servicio de Amazon Redshift fue utilizado para implementar el esquema de estrella que se puede observar en la Figura 12. En este se almacenan los datos que serán consumidos por las aplicaciones de BI.

A continuación, se presentan las estructuras de datos de las tablas creadas en Amazon Redshift.

Redshift query editor v2

addressdimension

	Field	Type	NL	CMP
#	addresskey	integer	NN	az64
#	addressbk	integer	NN	az64
#	customerbk	integer	NN	az64
A	country	character varying(256)	NN	lzo
A	state	character varying(256)	NN	lzo
A	company	character varying(256)	NN	lzo
A	address	character varying(256)	NN	lzo
A	postalcode	character varying(256)	NN	lzo
A	city	character varying(256)	NN	lzo
A	phone	character varying(256)	NN	lzo
A	phonemobile	character varying(256)	NN	lzo
A	deleted	character varying(256)	NN	lzo

Figura 26: Estructura de datos de la dimensión «addressdimension»

Redshift query editor v2

customerdimension

	Field	Type	NL	CMP
#	customerkey	integer	NN	az64
#	customerbk	integer	NN	az64
A	firstname	character varying(256)	NN	lzo
A	lastname	character varying(256)	NN	lzo
📅	birthday	date	NN	az64
A	gender	character varying(256)	NN	lzo
A	email	character varying(256)	NN	lzo
#	isguest	integer	NN	az64
📅	dateadd	date	NN	az64

Figura 27: Estructura de datos de la dimensión «customerdimension»

Redshift query editor v2

productdimension

	Field	Type	NL	CMP
#	productkey	integer	NN	az64
#	productbk	integer	NN	az64
A	productname	character varying(256)	NN	lzo
#	price	numeric(18,0)	NN	az64
A	category	character varying(256)	NN	lzo
A	brandname	character varying(256)	NN	lzo
A	attributes	character varying(256)	NN	lzo
A	features	character varying(256)	NN	lzo

Figura 28: Estructura de datos de la dimensión «productdimension»

Redshift query editor v2

salesfacttable

	Field	Type	NL	CMP
#	orderkey	integer	NN	az64
#	orderbk	integer	NN	az64
#	customerbk	integer	NN	az64
#	shopbk	integer	NN	az64
#	datekey	integer	NN	az64
#	productbk	integer	NN	az64
A	payment	character varying(256)	NN	lzo
A	currentstate	character varying(256)	NN	lzo
#	productquantity	integer	NN	az64
#	unitpricetaxincl	numeric(18,0)	NN	az64
#	totalshippingtaxincl	numeric(18,0)	NN	az64
#	totaldiscounts	numeric(18,0)	NN	az64
#	totalpaid	numeric(18,0)	NN	az64
📅	dateadd	date	NN	az64
📅	dateupd	date	NN	az64

Figura 29: Estructura de datos de la tabla de hechos «salesfacttable»

Redshift query editor v2

shopdimension

	Field	Type	NL	CMP
#	shopkey	integer	NN	az64
#	shopbk	integer	NN	az64
A	shopname	character varying(256)	NN	lzo

Figura 30: Estructura de datos de la dimensión «shopdimension»

Redshift query editor v2

datedimension

	Field	Type	NL	CMP
#	datekey	integer	NN	az64
📅	fulldate	date	NN	az64
#	dayofweek	integer	NN	az64
#	daynuminmonth	integer	NN	az64
#	daynumoverall	integer	NN	az64
A	dayname	character varying(256)	NN	lzo
A	dayabbev	character varying(256)	NN	lzo
#	weekdayflag	integer	NN	az64
#	weeknuminyear	integer	NN	az64
#	weeknumoverall	integer	NN	az64
📅	weekbegindate	date	NN	az64
#	month	integer	NN	az64
#	monthnumoverall	integer	NN	az64
A	monthname	character varying(256)	NN	lzo
A	monthabbev	character varying(256)	NN	lzo
#	quarter	integer	NN	az64
#	year	integer	NN	az64
#	yearmo	integer	NN	az64
#	fiscalmonth	integer	NN	az64
#	fiscalquarter	integer	NN	az64
#	fiscalyear	integer	NN	az64
#	monthendflag	integer	NN	az64
📅	samedayyearago	date	NN	az64

Figura 31: Estructura de datos de la dimensión «datedimension»

Aplicaciones de BI

El software elegido para contestar a las métricas definidas mediante la elaboración de tableros es Power BI. Se utiliza una conexión directa con Amazon Redshift para consumir los datos necesarios para que los tableros funcionen. Se demuestra cómo funcionan en el capítulo III.

Descripción de la tecnología a utilizar

En esta sección se describirán las tecnologías que utilizamos para el desarrollo de la propuesta de solución.

Amazon S3

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Este servicio ofrece la capacidad de almacenar y proteger cualquier cantidad de datos para prácticamente cualquier caso de uso, como los lagos de datos, las aplicaciones nativas en la nube y las aplicaciones móviles. Gracias a las clases de almacenamiento rentables y a las características de administración fáciles de usar, es posible optimizar los costos, organizar los datos y configurar controles de acceso detallados para cumplir con requisitos empresariales, organizacionales y de conformidad específicos.

El servicio puede usarse para una gran cantidad de usos. Por ejemplo: Creación de un lago de datos, copia de seguridad y restauración de datos fundamentales, archivo de datos con el costo más bajo, ejecución de aplicaciones nativas en la nube, entre otros.

Algunas ventajas que ofrece Amazon S3 son:

- Escalabilidad
- Capacidad de almacenamiento infinita: llegará dónde se esté dispuesto a pagar
- Integración completa con otros servicios de AWS

Amazon Redshift

Amazon Redshift utiliza SQL para analizar datos estructurados y semiestructurados en almacenamientos de datos, bases de datos operativas y lagos de

datos, con hardware y machine learning diseñado por AWS para ofrecer rendimiento al mejor precio a cualquier escala.

El servicio puede usarse para una gran cantidad de usos. Por ejemplo: Mejorar pronósticos financieros y de demanda, colaborar y compartir datos, optimizar la inteligencia empresarial, aumentar la productividad de los desarrolladores, entre otros.

Algunas ventajas que ofrece Amazon Redshift son:

- Almacenamiento en caché de resultados
- Integración completa con otros servicios de AWS
- Ofrece copias de seguridad coherentes de los datos

Talend Open Studio

Talend Open Studio (TOS) es una suite que aporta un conjunto muy complejo, variado y completo de herramientas para llevar a cabo la integración de datos que se ofrece en una versión de código libre. Es una de las herramientas de integración ETL (extract, transform, load) más utilizadas dentro del mundo Big Data; es más, es la cuarta en la lista después de Informática Powercenter, IBM InfoSphere Datastage y Oracle Data Integrator (ODI).

Algunas ventajas que ofrece Talend son:

- Es un all-in-one, es decir, que permite reducir el número de herramientas y configuraciones adicionales
- Tiene una amplia comunidad y mucha documentación
- Talend provee su propio framework para desarrollar componentes personalizados

Microsoft Power BI

Microsoft Power BI es un servicio de análisis de datos de Microsoft orientado a proporcionar visualizaciones interactivas y capacidades de inteligencia empresarial con una interfaz lo suficientemente simple como para que los usuarios finales puedan crear por sí mismos sus propios informes y paneles.

Algunas ventajas que ofrece Microsoft Power BI son:

- Fácil de usar
- Es asequible
- No tiene restricciones de memoria

PrestaShop

PrestaShop es un sistema de gestión de contenidos (CMS) libre y de código abierto pensado para construir desde cero tiendas en línea de comercio electrónico. Este es el sistema origen de donde se obtuvieron los datos.

Algunas ventajas que PrestaShop ofrece son:

- Es de código abierto y gratuito
- Escalable
- Fácil de utilizar

MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

Diagrama arquitectónico de la solución

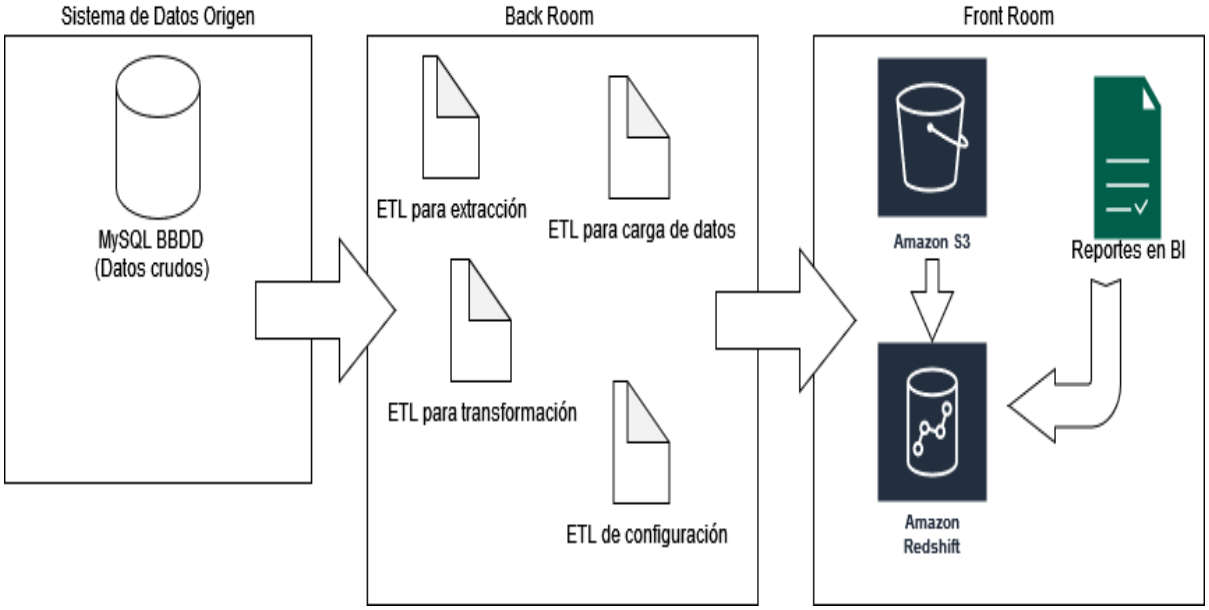

Figura 32: Diagrama arquitectónico de la solución

Descripción de cada componente de la solución

En esta sección se detalla en qué forma se utilizaron las tecnologías descritas en el apartado anterior.

PrestaShop

Usado únicamente para la generación de datos de prueba, realizando transacciones de registro de datos de clientes, productos y realizando compras.

MySQL

Usado en consecuencia de usar PrestaShop, ya que es la base de datos con la que trabaja este CMS. Se le dio uso para realizar el análisis de los esquemas a emplear, para verificar la calidad de datos y de analizar la definición de cada tabla y tipo de dato en ella para cada campo a extraer mediante los procesos ETL, además de formar parte de la solución la creación de una base de datos que almacena tablas para la configuración de los procesos etl y el registro de fallos que pudieran producir los procesos ETL.

Talend Open Studio

Herramienta empleada para el desarrollo de los procesos ETL que llevan a cabo la extracción de los datos necesarios para llenar el modelo dimensional creado, realizan la limpieza de datos a modo de llevar estos últimos al formato y tipo que necesita el modelo dimensional y cargan los datos en el área de presentación (S3 y Amazon Redshift).

Amazon S3

Componente empleado como lago de datos, para almacenar los datos procesados mediante los ETL desarrollados. En este se almacenan los datos nuevos y datos actualizados y de este punto son cargados a Amazon Redshift para ser consumidos por aplicaciones de BI.

Amazon Redshift

Componente de Amazon empleado para definir el modelo dimensional creado y para almacenar los datos extraídos mediante los procesos ETL.

Aplicaciones de BI

Los reportes construidos con la herramienta Power BI, consumen los datos de Amazon Redshift, para dar respuesta a las métricas definidas por los usuarios de negocio.

Estrategia de Implementación

Estrategia de implementación

A continuación, se propone una estrategia de implementación que permita a la empresa ejecutar la propuesta de solución. Cabe destacar que se trata de un producto nuevo, ya que la empresa no contaba con ninguna solución previa a la realización de esta propuesta.

Los pasos propuestos para poner en marcha la propuesta de solución son:

1. Dado que el servidor de la empresa usa el CMS de PrestaShop, por ende, no es necesario realizar la instalación de MySQL dado que este CMS lo utiliza por defecto, entonces se procede a crear una base de datos y dos tablas con las siguientes características:

   Nombre de la base de datos: distribuidora_el_aguila_configs

   Tablas a crear: jobs_config, jobs_exec_logs

   Sentencia DDL para la creación de la tabla jobs_config

```
CREATE TABLE `jobs_config` (
`id` bigint(20) UNSIGNED NOT NULL,
`key` varchar(150) NOT NULL,
`value` varchar(255) NOT NULL,
`date_add` date DEFAULT current_timestamp(),
`date_upd` date DEFAULT current_timestamp()
);
ALTER TABLE `jobs_config` ADD PRIMARY KEY (`id`);
ALTER TABLE `jobs_config` MODIFY `id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=26;
COMMIT;
```

   Sentencias DML para creación de registros necesarios para el correcto funcionamiento de los procesos etl.

```
INSERT INTO `jobs_config` (`id`, `key`, `value`, `date_add`, `date_upd`) VALUES
```

```
    (1, 'CustomerDimension', '0', '2022-12-19', '2022-12-
19'),
    (2, 'AddressDimension', '0', '2022-12-19', '2022-12-
19'),
    (3, 'ProductDimension', '0', '2022-12-19', '2022-12-
19'),
    (4, 'ShopDimension', '0', '2022-12-19', '2022-12-19'),
    (5, 'ServerFilePath', 'C:/ruta_base/', '2022-12-20',
'2022-12-20'),
    (6, 'CustomerDimFileName', 'CustomerDimension.csv',
'2022-12-20', '2022-12-20'),
    (7, 'AddressDimFileName', 'AddressDimension.csv',
'2022-12-20', '2022-12-20'),
    (8, 'ProductDimFileName', 'ProductDimension.csv',
'2022-12-20', '2022-12-20'),
    (9, 'ShopDimFileName', 'ShopDimension.csv', '2022-12-
20', '2022-12-20'),
    (10, 'RawFolderName', 'raw/', '2022-12-23', '2022-12-
23'),
    (11, 'StagingFolderName', 'staging/', '2022-12-23',
'2022-12-23'),
    (12, 'PresentationFolderName', 'presentation/', '2022-
12-23', '2022-12-23'),
    (13, 'AWSAccessKeyId', 'AKIARRBPW2TGJWOZH75L', '2022-
12-24', '2022-12-24'),
    (14, 'AWSSecretKey', 'HzERboFSIQDDyADOGjqcm7BdluJ7',
'2022-12-24', '2022-12-24'),
    (15, 'BucketName', 'files-el-aguila', '2022-12-24',
'2022-12-24'),
    (16, 'S3FolderName', 'presentation/', '2022-12-24',
'2022-12-24'),
    (17, 'NewDataFolder', 'new-data/', '2022-12-24',
'2022-12-24'),
```

```
    (18, 'UpdateDataFolder', 'updated-data/', '2022-12-24', '2022-12-24'),
    (19, 'LastJobExecDateAddress', '1000-01-01 00:00:00', '2022-12-26', '2022-12-26'),
    (20, 'LastJobExecDateCustomer', '1000-01-01 00:00:00', '2022-12-26', '2022-12-26'),
    (21, 'LastJobExecDateShop', '1000-01-01 00:00:00', '2022-12-26', '2022-12-26'),
    (22, 'LastJobExecDateProduct', '1000-01-01 00:00:00', '2022-12-26', '2022-12-26'),
    (23, 'OrdersFactFileName', 'OrdersFact.csv', '2023-01-01', '2023-01-01'),
    (24, 'LastJobExecDateOrders', '1000-01-01 00:00:00', '2023-01-01', '2023-01-01'),
    (25, 'OrdersDimension', '0', '2023-01-07', '2023-01-07');
```

Sentencia DDL para la creación de la tabla jobs_exec_logs

```
CREATE TABLE `jobs_exec_logs` (
  `moment` datetime NOT NULL,
  `pid` varchar(20) NOT NULL,
  `root_pid` varchar(20) NOT NULL,
  `father_pid` varchar(20) NOT NULL,
  `project` varchar(50) NOT NULL,
  `job` varchar(255) NOT NULL,
  `context` varchar(50) NOT NULL,
  `priority` int(3) NOT NULL,
  `type` varchar(255) NOT NULL,
  `origin` varchar(255) NOT NULL,
  `message` varchar(255) NOT NULL,
  `code` int(3) NOT NULL
);
```

2. Crear una tarea programada en el servidor para le ejecución del proceso etl en el período de tiempo que se desea realizar las tareas de extracción, transformación y carga de datos, primero que todo se debe tener en el servidor el proyecto de procesos etl de manera autónoma, y solo se debe programar la tarea de la siguiente manera:

   Los archivos crontab se ubican en la ruta **/var/spool/cron/crontabs, para programar la tarea con el usuario deseado, en el servidor, ejecutar**

   crontab -e

   Definir en el archivo el script del proceso etl autónomo a ejecutar en el período de tiempo que se desea, para programarlo tomar en cuenta el siguiente formato

   “minutos” “horas” “día del mes” “día de la semana” “usuario” “comando o script”

   Ejemplo: 55 23 * * 0 root /home/usuario/script1.sh

3. Definir en S3 un bucket, con la configuración deseada para el almacenamiento de los archivos CSV, simplemente accediendo con una cuenta en AWS y buscando el componente S3, y configurar el bucket con el almacenamiento deseado.

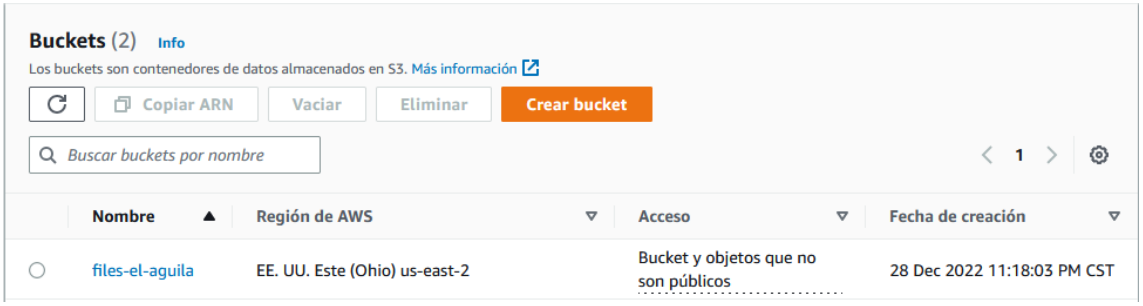

Figura 33: Creación de bucket en Amazon S3

Generar un par de claves de acceso, para su almacenamiento en las tablas de configuración (tabla jobs_config), específicamente en la columna «value» en los registros con los campos

Campo ID -> 13, Campo key -> 'AWSAccessKeyId'

Campo ID -> 14, Campo key -> 'AWSSecretKey'

Campo ID -> 15, Campo key -> 'BucketName'

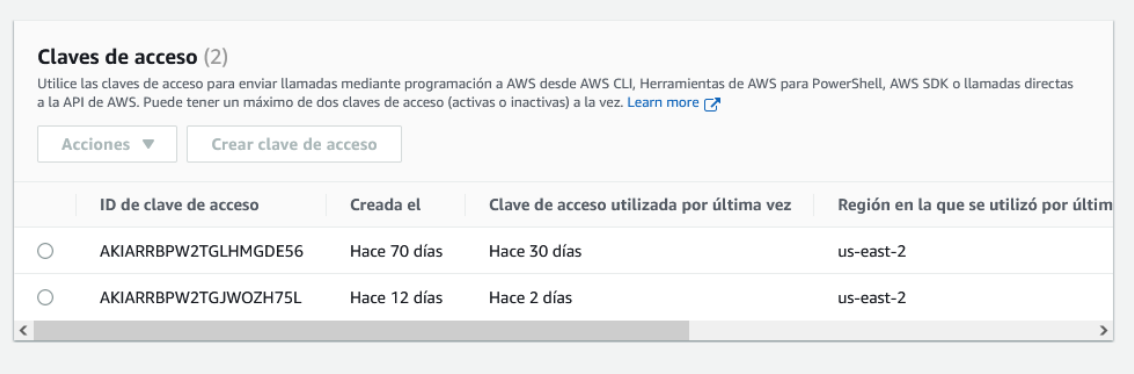

Figura 34: Generación de claves de acceso

En la imagen, se puede ver la sección para crear las claves de acceso, estas son las que deben colocarse en los registros antes mencionados junto con el nombre del bucket.

4. Definir el esquema estrella en Amazon Redshift, ingresar a la consola de Amazon Redshift, crear un esquema nuevo o una base de datos y definir ejecutar los siguientes scripts de creación de tablas

```
create table schema_name.datedimension(
    DateKey integer not null,
    FullDate date not null,
    DayOfWeek integer not null,
    DayNumInMonth integer not null,
    DayNumOverall integer not null,
    DayName varchar not null,
    DayAbbev varchar not null,
    WeekdayFlag integer not null,
    WeekNumInYear integer not null,
    WeekNumOverall integer not null,
    WeekBeginDate date not null,
    Month integer not null,
    MonthNumOverall integer not null,
    MonthName varchar not null,
    MonthAbbev varchar not null,
    Quarter integer not null,
    Year integer not null,
    YearMo integer not null,
```

```
        FiscalMonth integer not null,
        FiscalQuarter integer not null,
        FiscalYear integer not null,
        MonthEndFlag integer not null,
        SameDayYearAgo date not null,
        primary key(DateKey)
    )
    CREATE TABLE schema_name.customerdimension(
    customerkey integer NOT NULL encode az64,
         customerbk  integer NOT NULL encode az64,
         firstname   character varying(256) NOT NULL encode
lzo,
         lastname    character varying(256) NOT NULL encode
lzo,
         birthday    date NOT NULL encode az64,
         gender     character varying(256) NOT NULL encode
lzo,
         email      character varying(256) NOT NULL encode
lzo,
         isguest     integer NOT NULL encode az64,
         dateadd     date NOT NULL encode az64,
    CONSTRAINT        customerdimension_pkey        PRIMARY
KEY(customerkey));
    CREATE TABLE schema_name.addressdimension(
         addresskey  integer NOT NULL encode az64,
         addressbk   integer NOT NULL encode az64,
         customerbk  integer NOT NULL encode az64,
         country    character varying(256) NOT NULL encode
lzo,
         state      character varying(256) NOT NULL encode
lzo,
         company    character varying(256) NOT NULL encode
lzo,
```

```
          address    character varying(256) NOT NULL encode
lzo,
          postalcode  character varying(256) NOT NULL encode
lzo,
          city       character varying(256) NOT NULL encode
lzo,
          phone      character varying(256) NOT NULL encode
lzo,
          phonemobile   character   varying(256)   NOT   NULL
encode lzo,
          deleted    character varying(256) NOT NULL encode
lzo,
     CONSTRAINT        addressdimension_pkey        PRIMARY
KEY(addresskey));
     CREATE TABLE schema_name.productdimension(
          productkey  integer NOT NULL encode az64,
          productbk   integer NOT NULL encode az64,
          productname   character   varying(256)   NOT   NULL
encode lzo,
          price       numeric(18,0) NOT NULL encode az64,
          category    character varying(256) NOT NULL encode
lzo,
          brandname   character varying(256) NOT NULL encode
lzo,
          attributes  character varying(256) NOT NULL encode
lzo,
          features    character varying(256) NOT NULL encode
lzo,
     CONSTRAINT        productdimension_pkey        PRIMARY
KEY(productkey));

     CREATE TABLE schema_name.salesfacttable(
          orderkey            integer NOT NULL encode az64,
```

```
            orderbk             integer NOT NULL encode az64,
            customerbk          integer NOT NULL encode az64,
            shopbk              integer NOT NULL encode az64,
            datekey             integer NOT NULL encode az64,
            productbk           integer NOT NULL encode az64,
            payment    character varying(256) NOT NULL encode
lzo,
         currentstate  character  varying(256)  NOT  NULL
encode lzo,
         productquantity     integer NOT NULL encode az64,
         unitpricetaxincl    numeric(18,0) NOT NULL encode
az64,
         totalshippingtaxincl numeric(18,0) NOT NULL encode
az64,
         totaldiscounts      numeric(18,0) NOT NULL encode
az64,
         totalpaid           numeric(18,0) NOT NULL encode
az64,
         dateadd             date NOT NULL encode az64,
         dateupd             date NOT NULL encode az64,
     CONSTRAINT         salesfacttable_pkey         PRIMARY
KEY(orderkey,orderbk,customerbk,shopbk,datekey,productbk);
     CREATE TABLE schema_name.shopdimension(
           shopkey  integer NOT NULL encode az64,
           shopbk   integer NOT NULL encode az64,
           shopname  character  varying(256)  NOT  NULL  encode
lzo,
     CONSTRAINT shopdimension_pkey PRIMARY KEY(shopkey));
```

5. Realizar la conexión a Redshift con Power BI para la creación de tableros o para su visualización

Presupuesto de implementación

En esta sección se presenta el presupuesto de implementación que necesitaría la empresa Distribuidora El Águila para poder implementar la propuesta de solución. Se consideran todos los recursos necesarios para una correcta implementación de la solución. Se incurriría en recurso humano, tecnológico, entre otros.

Se proponen los costos mensuales para cada recurso.

Recurso humano

Tomando como base el salario de ingeniero de datos ($1300/mes) y el de un analista de datos ($914/mes)

Recurso humano	Cantidad	Salario mensual ($)	Total ($)
Ingeniero de datos	2	1,300	2,600
Analista de datos	1	950	950
Total			3,550

Cabe recalcar que los ingenieros de datos se encargarían de montar la solución en los servicios en la nube y posteriormente, darían mantenimiento a la arquitectura de data warehouse propuesta. Por otro lado, se propone la contratación de un analista de datos que será el encargado de generar los tableros en Power BI, analizar las tendencias y auxiliar con información útil a los tomadores de decisiones de la empresa.

Recurso tecnológico

El recurso tecnológico para la implementación no contemplaría un servidor local, sino que se enfocaría en contratar los servicios de Amazon AWS para su implementación. En el caso de la propuesta de solución, se plantea contratar los servicios de Amazon S3 y Amazon Redshift. Con respecto a Talend Open Studio, se propone utilizar la capa gratuita debido a que, con esta capa, ya se incluyen todos los servicios necesarios para implementar la solución.

El volumen de datos estimado en el primer año es de 160 GB, y se estima que cada año ese volumen de datos aumente en la misma cantidad, se estima que en cinco años, el total de información almacenada en Amazon S3 sea de 800 GB.

Servicio	**Precios de almacenamiento ($/GB)**	**Almacenamiento estimado en primer año (GB)**	**Almacenamiento estimado en segundo año (GB)**	**Almacenamiento estimado en tercer año (GB)**	**Almacenamiento estimado en cuarto año (GB)**	**Almacenamiento estimado en quinto año (GB)**	**Total en 5 años ($)**
Amazon S3 Standard	0.026	160	320	480	640	800	748.80
Total							748.80

Los costos mensuales serían la cantidad de GB reservados en S3 por el precio de almacenamiento, por lo que este costo va a variar mes a mes a medida que se ocupe más almacenamiento. Suponiendo que la solución esté activa por cinco años y que el almacenamiento utilizado aumente en 160 GB por año, el costo total por los cinco años sería de $748.80.

Se estima que el precio bajo demanda que ofrece Amazon Redshift Informática densa DC2, con 2 CPU virtuales, 15 GiB de memoria, 0.16 TB SSD de capacidad de almacenamiento a disposición, y 0.60 GB/s de E/S es suficiente para implementar la solución en Amazon Redshift.

Servicio	**Precio ($/hora)**	**Tiempo de uso (horas/día)**	**Total al mes ($)**
Amazon Redshift DC2.large	0.33	8	79.20
Total			79.20

Para los costos de Amazon Redshift, se estiman costos mensuales de $79.20. Esto debido a que solo se toman en cuenta ocho horas de uso al día por parte de los usuarios del negocio.

Suponiendo que la cantidad de usuarios del negocio a utilizar Power BI es de 10, se propone utilizar la versión Pro de este último.

Servicio	Costo (usuario/mes)	Total mensual ($)
Power BI	9.99	99.9
Total		99.9

Con base en lo anterior, ahora se calculan los costos totales mensuales.

Recurso	Total ($)
Humano	3,550
Amazon S3	4.16
Amazon Redshift	79.20
Power BI	99.9
Total	3,733.26

El total mensual estimado para la implementación de la propuesta de solución sería de $3,733.26 por mes. Este costo variará con el tiempo a medida se aumente el almacenamiento en los componentes en la nube. Además, el costo puede verse incrementado debido a la cantidad de operaciones de lectura o escritura que se realicen contra estos componentes al mes.

Análisis de resultados

En esta sección se presentan las evidencias más destacadas de los resultados obtenidos de la propuesta de solución. Para este caso, las evidencias son las capturas de los tableros construidos en Power BI a partir de la información obtenida mediante la solución. Cada tablero representa una de las métricas establecidas.

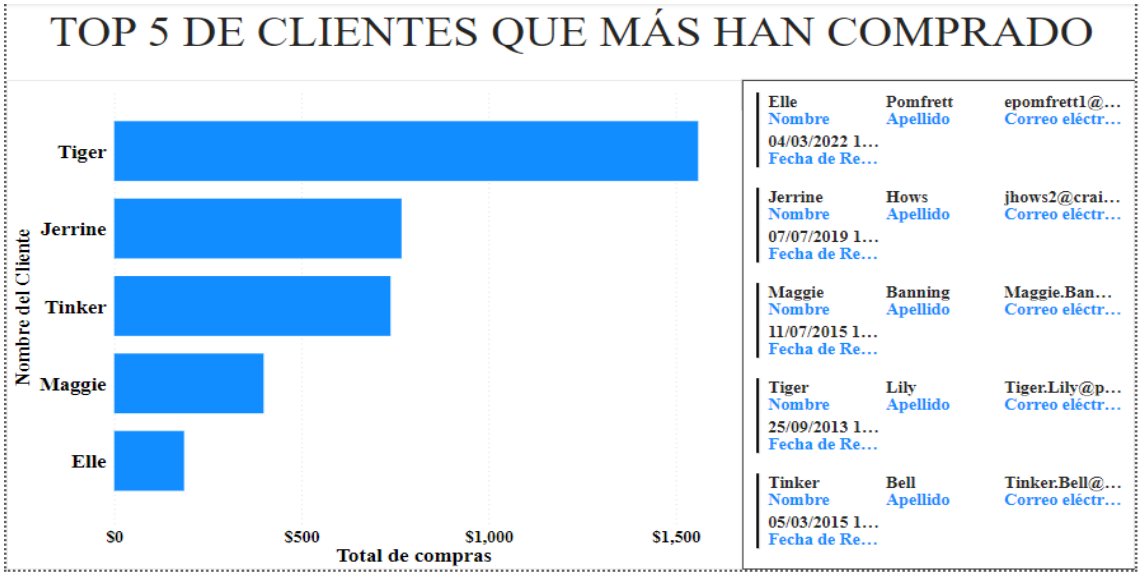

Figura 35: Tablero con el top 5 de los clientes que más han comprado

Este tablero muestra el top 5 de los clientes que más han comprado (monto en $) ya sea en la tienda física o en la tienda en línea. Los tomadores de decisiones podrían basarse en esta información para decidir qué clientes pueden aplicar a ciertas promociones especiales o a qué clientes les pueden ofrecer ciertos créditos.

Figura 36: Tablero que muestra el top 5 de productos que más se han vendido

En este tablero se puede observar el top 5 de productos que más se han vendido (monto en $) ya sea en la tienda física o en la tienda en línea. Los tomadores de decisiones podrían utilizar esta información para decidir qué productos no necesitan tanta promoción o para ajustar sus estrategias de venta para vender más ciertos productos.

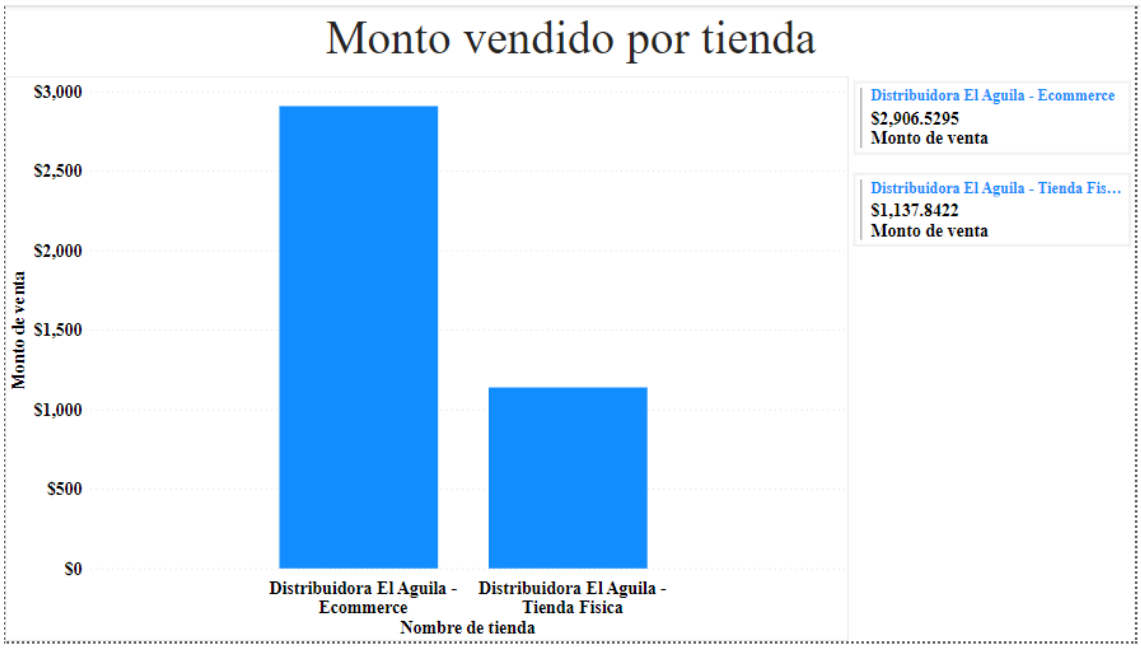

Figura 37: Tablero que muestra el monto vendido por tienda

En este tablero se puede observar el monto vendido por tienda, ya sea la tienda física o la tienda en línea. Con esta información, los tomadores de decisiones podrían ajustar sus estrategias para promocionar más alguna de sus sucursales. En caso no se esté vendiendo mucho en la tienda física, se podría promocionar más o dar más soporte a la tienda en línea ya que esta se utiliza más.

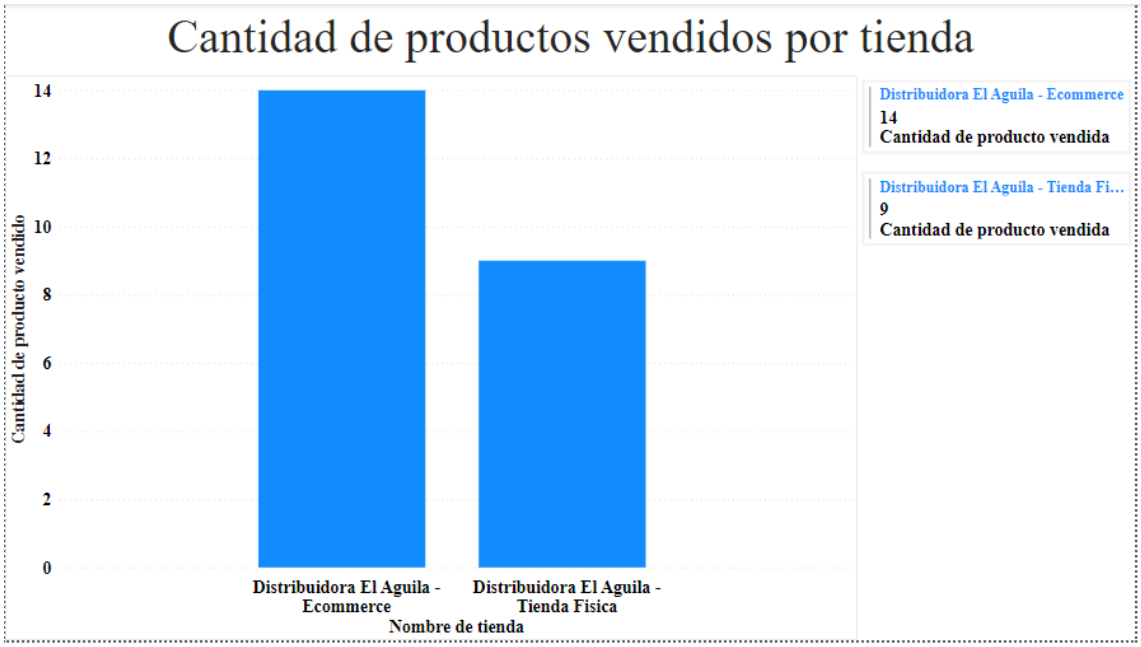

Figura 38: Tablero que muestra la cantidad de productos vendidos por tienda

Este tablero muestra la cantidad de productos vendidos por tienda, ya sea la tienda física o la tienda en línea. Esta información muestra qué tienda vende más productos.

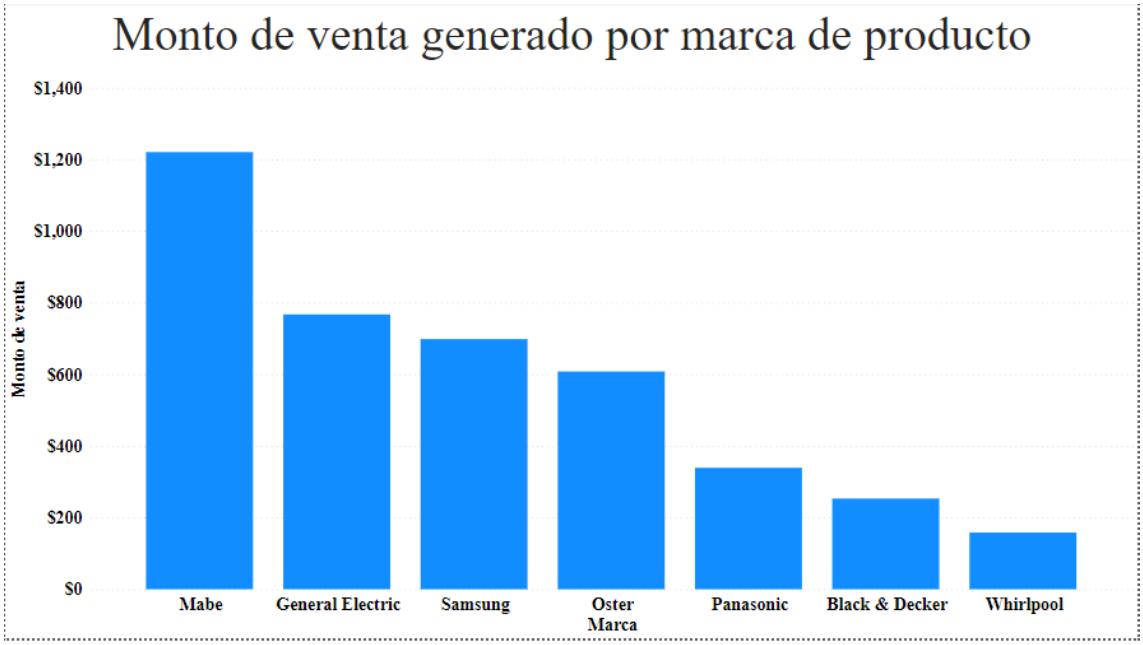

Figura 39: Tablero que muestra el monto de venta generado por marca de producto

Este tablero muestra el monto de venta generado por marca de producto. Con esta información, los tomadores de decisiones podrían conocer qué marcas generan más dinero.

Conclusiones y recomendaciones

Conclusiones

En conclusión, se lograron los objetivos establecidos en este proyecto de investigación y análisis de un modelo de arquitectura de data warehouse

para la empresa Distribuidora El Águila. Se realizó un análisis exhaustivo de la situación actual de la empresa, sus esquemas de base de datos, y se diseñó una propuesta de arquitectura de data warehouse que permita el análisis de los datos del área de ventas para la toma de decisiones basadas en información.

La propuesta de solución fue construida garantizando la calidad y presentación de los datos, y se diseñó un plan de implementación para llevar a cabo la propuesta de manera efectiva y eficiente. De esta manera, se espera que esta propuesta de arquitectura de data warehouse ayude a la empresa Distribuidora El Águila a mejorar sus procesos de toma de decisiones y aumentar su competitividad en el mercado.

Recomendaciones

Para este tipo de proyecto, se recomienda tener una buena comunicación con el cliente final, debido a que los servicios más importantes son servicios en la nube, y tener una buena comunicación permitirá que se escojan los servicios que más se acoplen a las necesidades de los usuarios y a su presupuesto.

Además, es importante definir de manera precisa y sin ambigüedades, los requerimientos o métricas que se desean responder con la solución. Esto debido a que, con base a las métricas definidas, así serán las necesidades que presente la solución.

Para la implementación de la solución en Amazon S3, se recomienda tener en consideración las siguientes opciones: Configurar las Access Control Lists (ACL) o Listas de Control de Acceso para controlar quién tiene y quién no tiene acceso a los buckets; también se recomienda configurar el servicio AWS Identity and Access Management (IAM) o Gestión de Identidad y Acceso para controlar los accesos a los recursos de AWS en general, y para controlar quién puede autenticarse en una cuenta y quién puede utilizar recursos; además se recomienda no utilizar el usuario «root», y en su lugar, se recomienda crear usuarios IAM y asignar explícitamente los recursos a los que tendrán acceso; para finalizar, se recomienda seleccionar la región más cercana a la zona en donde se consumirán los datos, en nuestro caso se recomienda elegir la zona oeste de los Estados Unidos.

Por otro lado, se recomienda que los datos se consuman localmente, es decir, en El Salvador, debido a que la Ley Salvadoreña no permite que los datos sean consumidos fuera del país, pero sí pueden ser almacenados fuera.

Bibliografía

- 23 pros y contras de Amazon Redshift. (s/f). Paginapropia.com. Recuperado enero de 2023, de https://paginapropia.com/23-pros-y-contras-de- amazon-redshift/
- Amazon. (s/f-a). Amazon.com. Recuperado enero de 2023, de https://aws.amazon.com/es/s3/
- Amazon. (s/f-b). Amazon.com. Recuperado enero de 2023, de https://aws.amazon.com/es/redshift/
- Ejemplos de Justificación (de un proyecto o investigación). (s/f). Ejemplos.co. Recuperado enero de 2023, de https://www.ejemplos.co/7-ejemplos-de-justificacion-de-trabajo-o-investigacion/
- FreshCommerce, E. (2022, julio 13). Qué es Talend. Pros y Contras. FreshCommerce. https://freshcommerce.es/blog/herramientas-etl-talend/
- Latinoamérica, S. (s/f). Departamento de Ventas: Cuáles son sus funciones. Blog de Salesforce. Recuperado enero de 2023, de https://www.salesforce.com/mx/blog/2021/06/departamento-de-ventas-cuales-son-sus-funciones.html
- Portal, T. I. C. (2015, noviembre 10). Amazon S3 y el almacenamiento en la nube de Amazon. TIC Portal. https://www.ticportal.es/temas/cloud-computing/amazon-web-services/amazon-s3
- Publicado por pmoinformatica.com. (s/f). Pruebas de caja negra ISTQB. Pmoinformatica.com. Recuperado enero de 2023, de http://www.pmoinformatica.com/2016/04/pruebas-caja-negra-istqb.html
- ¿Qué es Talend Open Studio? (2022, mayo 23). KeepCoding Tech School. https://keepcoding.io/blog/talend-open-studio/

- Santiago, I. (2022, abril 26). 6 ventajas de usar Power BI. Instituto Cibertec | Carreras Técnicas y Formación Continua; Cibertec. https://www.cibertec.edu.pe/noticias/cuales-son-las-ventajas-de-usar-power-bi/