

DASHBOARD PARA LA VISUALIZACIÓN DE INDICADORES SOBRE EL USO QUE TIENEN LOS CLIENTES DEL APLICATIVO WEB SICEX

Autores

JACKELINE PINEDA MOLINA
EDWIN LEONARDO FUENTES DAZA

Director

Elio H. Cables Pérez, Ph.D

Universidad Antonio Nariño
Facultad de Ingeniería de Sistemas
Especialización en Gobierno de Datos
Bogotá D.C
2023

Índice de contenidos

1. Introducción	1
2. Descripción y formulación del problema	2
2.1 Objetivo General	3
2.2 Objetivos Específicos	3
3. Marco referencial	4
3.1 Marco teórico	4
3.2 Estado del Arte	7
3.3 Impacto	8
3.4 Componente de Innovación	8
4. Metodología	9
5. Desarrollo de la propuesta	13
5.1 Análisis de los requisitos	13
5.2 Modelo Dimensional	14
5.3 Diseño físico	16
5.3.1 Diseño del Modelo Físico	16
5.3.2 Implementación del modelo Físico	17
5.3.3 Diccionario de Datos:	17
5.4 Diseño ETL	17
5.4.1 Monitoreo de finalización del proceso ETL	18
5.4.2 Validación información cargada	¡Error! Marcador no definido.
5.5 Especificaciones de aplicación BI	26
5.5.1 Análisis de caso de uso	26
5.5.2 Definición de métricas clave del Dashboard	26
5.5.3 Acceso al Dashboard y Clasificación de Usuarios	28
5.6 Desarrollo de aplicación BI	29
6. Conclusiones	36
7. Recomendaciones	37
8. Referencias	38
Anexos	1
Anexo 1 Diccionario de datos	1
Anexo 2 Script creación bodega de datos	4

Anexo 3 ConexionBD.py 10
Anexo 4 ConexionCorreo.py 11
Anexo 5 ETL.py..... 12

Índice de Figuras

Figura 1 Arquitectura Data Warehouse Kimball	9
Figura 2 Metodología Kimball para el desarrollo del proyecto	11
Figura 3 Arquitectura Datamart	13
Figura 4 Modelo Físico Bodega de Datos.....	16
Figura 6 Evidencia Envío Correo Cuando el Proceso ETL no es Exitoso.....	18
Figura 7 Evidencia Envío Correo Cuando el Proceso ETL es Exitoso	19
Figura 8 Mensaje Confirmación Proceso ETL Exitoso	19
Figura 9 Mensaje Confirmación Proceso ETL no Exitoso.....	19
Figura 10 Validación Data Tabla dimCalendario.....	20
Figura 11 Validación Data Tabla dimAccion.....	20
Figura 12 Validación Data Tabla dimProducto	21
Figura 13 Validación Data Tabla dimIntercambio	21
Figura 14 Validación Data Tabla dimPais.....	22
Figura 15 Validación Data Tabla dimEmpresa	22
Figura 16 Validación Data Tabla dimUsuario.....	23
Figura 17 Validación Data Tabla dimCapitulo.....	23
Figura 18 Validación Data Tabla dimPartida	24
Figura 19 Validación Data Tabla dimSubpartida.....	24
Figura 20 Validación Data Tabla monitoreoAplicacion	25
Figura 21 Validación Data Tabla sesionAplicacion	25
Figura 22 Power BI Conexión a Bodega de datos	29
Figura 23 Power BI Estructura de Tablas	29
Figura 24 Dashboard General	30
Figura 25 Power BI Participación Según Empresa Cliente.....	31
Figura 26 Power BI Empresa Cliente Top 10	31
Figura 27 Power BI Participación Países	32
Figura 28 Power BI Top 10 Participación Países	32
Figura 29 Power BI Participación Producto	33
Figura 30 Power BI Participación Producto	33

Índice de tablas

Tabla 1 Métricas clave del Dashboard 26

Resumen

La empresa Quintero Hermanos SAS posee un valioso aplicativo web de consulta de información de comercio exterior. A pesar de que este aplicativo registra las actividades de los usuarios en un log, los datos contenidos en él no se utilizan eficazmente para comprender el uso de la aplicación por parte de los clientes.

Por otra parte, el área comercial carece de acceso a esta información, lo que limita su capacidad para brindar un asesoramiento efectivo a los clientes. El problema planteado se centra en cómo aprovechar los registros históricos generados por el uso del aplicativo web para mejorar el asesoramiento del área comercial a los clientes que realizan investigaciones de mercado.

El objetivo general de este proyecto es desarrollar un Dashboard que visualice indicadores sobre el uso del aplicativo web SICEX, brindando información detallada al equipo comercial para optimizar sus estrategias.

Palabras clave: Dashboard, Log, aplicativo web, comercio exterior.

Abstract

The company Quintero Hermanos SAS has a valuable web application for consulting foreign trade information. Although this application records user activities in a log, the data contained therein is not used effectively to understand customers' use of the application.

On the other hand, the commercial area lacks access to this information, which limits its ability to provide effective advice to clients. The problem posed focuses on how to take advantage of the historical records generated by the use of the web application to improve the advice of the commercial area to clients who carry out market research.

The general objective of this project is to develop a Dashboard that displays indicators on the use of the SICEX web application, providing detailed information to the commercial team to optimize their strategies.

Keywords: Dashboard, Log, web application, foreign trade.

1. Introducción

En el ámbito del comercio exterior, la empresa Quintero Hermanos SAS, con una vasta experiencia de 100 años en la prestación de servicios de investigación de mercados, se ha destacado por suministrar información de importaciones y exportaciones de Colombia y otros países.

Como parte de sus recursos clave, la empresa ofrece un valioso aplicativo web que permite a sus clientes acceder a información detallada sobre el comercio exterior en Colombia. Este aplicativo brinda diversas funcionalidades, como consultas detalladas, creación de Dashboards y tablas dinámicas para análisis.

A pesar de contar con un registro exhaustivo de todas las actividades de los usuarios en la aplicación, esta valiosa información permanece en una tabla semiestructurada, de acceso exclusivo para el administrador de la base de datos. Como resultado, el equipo comercial de la empresa desconoce estas actividades y no puede comprender efectivamente el comportamiento y las preferencias de los usuarios. A pesar de almacenar registros de actividad, estos datos no se emplean de manera óptima para el beneficio del equipo comercial.

La relevancia de esta información se hace evidente, ya que podría permitir al área comercial visualizar las acciones de cada cliente y, en consecuencia, brindar un asesoramiento más informado y personalizado.

Este proyecto se enmarca en la necesidad del sector de comercio exterior de contar con información para evaluar y comprender su entorno de negocios. Al brindar una herramienta valiosa para el equipo comercial, se fortalecerá el asesoramiento a los clientes, lo que es crucial en un sector tan dinámico y competitivo.

En el desarrollo del proyecto, se detallarán los procesos y tecnologías utilizadas para lograr los objetivos propuestos y se presentan las conclusiones derivadas de la implementación de esta solución integral.

2. Descripción y formulación del problema

Actualmente la empresa Quintero Hermanos SAS presta servicios de investigación de mercados con 100 años de experiencia suministrando información de comercio exterior de las importaciones y exportaciones de Colombia y 36 países del mundo. La compañía cuenta con un valioso aplicativo web de consulta de información de comercio exterior SICEX (Sistema de información de comercio exterior), este permite realizar consultas sobre el comercio exterior en Colombia, el cual brinda varios recursos para el análisis a sus clientes como la consulta detallada, Dashboard y construcción de tablas dinámicas.

La base de datos contiene un log donde se registran todas las actividades que realiza el cliente en el aplicativo, este se encuentra en un campo de una tabla semiestructurada en un servidor de base de datos, encontrando allí información útil como el producto de selección, el país, el intercambio, el rango de fecha de consulta de la data y los filtros que se activaron para dicha consulta.

El acceso a esta información del log únicamente la tiene el administrador de la base de datos, por lo tanto, el área comercial la desconoce, y no existe un proceso que permita extraer los datos contenidos en él, lo que resulta en una falta de conocimiento por parte del equipo comercial sobre las preferencias y comportamientos de los usuarios. Aunque se almacenan registros de actividad, estos datos no se utilizan de manera efectiva para comprender el uso de la aplicación por parte de los clientes.

Por lo anterior, esta información es relevante para el área comercial ya que les permitiría visualizar las acciones que realiza cada uno de los clientes y tener la posibilidad de brindar un mejor asesoramiento a los clientes.

Formulación del problema

¿Cómo mejorar el asesoramiento del área comercial hacia los clientes que realizan investigación de mercados a partir de los registros históricos generados por el uso del aplicativo web para la consulta de información de comercio exterior en Colombia?

2.1 Objetivo General

Desarrollar un Dashboard para la visualización de indicadores sobre el uso que tienen los clientes del aplicativo web SICEX, que proporcione información detallada al equipo comercial permitiéndole la optimización de sus estrategias.

2.2 Objetivos Específicos

- Analizar la estructura y naturaleza de los registros de actividad almacenados en la base de datos semiestructurada de la aplicación web SICEX.
- Realizar un proceso ETL que permita la extracción eficiente de los datos no estructurados referentes al log de uso del aplicativo web SICEX.
- Crear una bodega de datos que almacene de manera estructurada la información del log de uso del aplicativo web SICEX.
- Implementar una visualización interactiva que permita al equipo comercial la exploración y análisis de los datos sobre el uso del aplicativo web SICEX por parte de los usuarios.

3. Marco referencial

El marco referencial de este trabajo proporciona un fundamento teórico y contextual para comprender la problemática abordada y la propuesta presentada. Se estructura en dos componentes fundamentales: el marco teórico y el estado del arte. El marco teórico se centra en los conceptos y teorías esenciales relacionados con el proyecto que se desarrolló, mientras que el estado del arte analiza el contexto de las empresas relacionadas con el tema, identificando las tendencias y avances relevantes en el ámbito del comercio exterior. Estos elementos permiten situar la contribución de este trabajo dentro del contexto académico y profesional en el que se desarrolla.

3.1 Marco teórico

El comercio exterior en Colombia es un componente fundamental de la economía y se refiere a todas las transacciones comerciales que involucran la compra y venta de bienes y servicios entre Colombia y otros países.

En el comercio exterior las agencias de aduanas siendo empresas autorizadas por la Dirección de Impuestos y Aduanas Nacionales (DIAN) para representar a los importadores y exportadores en los trámites aduaneros, actuando como intermediarios para gestionar la documentación y los procedimientos aduaneros en nombre de sus clientes, se aseguran de que se cumplan todas las regulaciones y requisitos, incluyendo la presentación de declaraciones de importación y exportación, ante la DIAN (DIAN, 2023).

La declaración de importación es un documento que debe presentar el importador ante la DIAN para informar sobre los bienes que está importando, su valor y otros detalles relevantes y la declaración de Exportación, en cambio, es un documento para informar sobre los bienes que está exportando, su valor y otros detalles. Así mismo, en el proceso de comercio exterior, se establece el arancel de aduanas que es un conjunto de tarifas o impuestos que se aplican a los bienes importados o exportados. La DIAN establece y regula estos aranceles, y las agencias de aduanas se encargan de calcular los costos aduaneros asociados a las transacciones comerciales (DIAN, 2023).

Por el proceso anterior, DIAN, es la entidad encargada de supervisar y regular las operaciones de comercio exterior en Colombia. Esta entidad establece las políticas aduaneras, recauda los impuestos de aduana y garantiza el cumplimiento de las regulaciones comerciales (DIAN, 2023).

Los puertos marítimos son puntos de entrada y salida para las mercancías en el comercio internacional. Es el lugar donde se realiza la inspección aduanera, se verifica la documentación y se llevan a cabo los procedimientos de importación y exportación. La información en los puertos marítimos es esencial para garantizar que las mercancías se muevan de manera eficiente y cumplan con todas las regulaciones aduaneras. Por esta razón, son clave en la información, porque es allí donde se llevan a cabo las operaciones físicas y documentales del comercio exterior.

El Departamento Administrativo Nacional de Estadística (DANE), Es una entidad encargada de recopilar, procesar y difundir información estadística en Colombia. Por lo tanto, recopila datos estadísticos relacionados con estas transacciones (importaciones y exportaciones), lo que es crucial para la elaboración de estadísticas económicas y el análisis de tendencias comerciales. En la DIAN también existen las Bases Estadísticas de Comercio Exterior, que “Comprenden el movimiento legal de mercancías que ingresan o salen del territorio aduanero nacional -TAN- a través de las diferentes aduanas del país, desde o hacia otros países y zonas francas del territorio nacional” (DANE, 2023).

SICEX al ser una plataforma online que suministra información de comercio exterior, con información oportuna y relevante de los movimientos diarios de carga marítima en los puertos colombianos, entiende que cada sector comercial es diferente y que la disponibilidad de información y servicios a ofrecer pueden ser parametrizables según criterios de análisis (SICEX, 2023).

Para el trabajo a realizar con el proyecto se gestionó grandes volúmenes de datos por lo que es necesario trabajar con bases de datos, las bases de datos son sistemas de almacenamiento de datos que permiten guardar y recuperar información de manera eficiente. (Microsoft, 2023)

Para realizar la gestión eficiente de las bases de datos se requiere de los sistemas de gestión de bases de datos o los gestores de bases de datos, que corresponde al software que administra y controla el acceso a la base de datos, particularmente en este proyecto se trabajó con MySQL, el cual es un motor de base de datos relacional, lo que permite organizar los datos en tablas que contienen filas y columnas, utilizando claves primarias y claves foráneas para crear relaciones entre las tablas, (MySQL). La versión de MySQL usada en este proyecto es 8.0.26 (MySQL, 2023).

Para obtener la información de la fuente de datos, se utilizó una ETL, que corresponde al proceso de extraer datos de múltiples fuentes, transformarlos para que sean coherentes y cargados en una base de datos. Una ETL “Es un proceso clave en la gestión de datos, y se refiere a las tres etapas fundamentales involucradas en la preparación y carga de datos en un almacén de datos o en

una base de datos para su posterior análisis. Las siglas ETL representan Extracción (Extract), Transformación (Transform) y Carga (Load)” (Pandey Brij, 2023)

Para trabajar con los procesos ETL, también es necesario articular otras herramientas como los lenguajes de programación. Existen diversos lenguajes de programación, sin embargo, por los requerimientos del proyecto se facilita el trabajo con Python el cual es un lenguaje de programación ampliamente utilizado en la manipulación y análisis de datos. (SAS, 2023)

Puede utilizarse para desarrollar scripts y aplicaciones que automatizan tareas de ETL, minería de datos y análisis de datos. Con todo el procesamiento a realizar a la información es necesario su almacenamiento, para ello son útiles las bodegas de datos que son un sistema de almacenamiento de datos diseñado para facilitar el análisis y la generación de informes (Pandey Brij, 2023).

Estas bodegas de datos brindan gran cúmulo de datos de forma general, sin embargo, se tienen que buscar mecanismos de resúmenes para poder interpretar de manera más fácil toda esa información. Esto lo proporcionan las herramientas de inteligencia de negocios, donde esta se refiere al proceso de recopilar, analizar y presentar información empresarial para facilitar la toma de decisiones informadas. (Oracle, 2023)

La herramienta usada en el presente proyecto es Power BI, la cual se utiliza para la creación de informes interactivos y paneles de control. Es una herramienta para el análisis empresarial que integra diversas fuentes de datos, obtener un análisis de estos por medio de visualizaciones que pueden compartirse a varios usuarios dentro de la misma empresa garantizando la disponibilidad de los reportes con información en tiempo real (Johnna, 2023).

El contexto de negocio y los anteriores conceptos técnicos se combinan para permitir la recopilación, integración, almacenamiento, análisis y visualización de datos del uso del aplicativo SICEX, en un tablero de control de inteligencia de negocios (BI) como Power BI. Esta integración y análisis de datos facilita la toma de decisiones para mejorar el nivel de asesoramiento hacia los clientes. (Powerdata, 2023)

3.2 Estado del Arte

La era digital ha cambiado la forma en que se interactúa con la información, donde los aplicativos webs de consulta han tomado un rumbo importante en su capacidad para generar registros de todas las actividades generadas por los usuarios. Estos registros llamados logs, encierran una riqueza de datos que adquieren una importancia estratégica, ya que es una puerta para comprender la interacción de los usuarios con los aplicativos web, identificando patrones de navegación y funciones más utilizadas que permiten a las empresas tener la posibilidad de mejorar la experiencia de los usuarios, tomar decisiones informadas, priorizar características, funcionalidades de acuerdo a la demanda y crear estrategias para la retención de clientes que conlleve a establecer un diferencial competitivo en un mercado dinámico y exigente.

Se analizaron diversos proyectos relacionados con el análisis de comportamiento de los usuarios en los aplicativos webs, y se encontró que el proyecto de la aplicación de Inteligencia empresarial para el análisis del comportamiento de incidencias de una plataforma de gestión de tarificación de usuario de operadoras de telefonía móvil se centra en analizar las incidencias en dicha plataforma para 16 operadoras en América y Europa. El proyecto siguió una metodología, en la que extrae datos de una base de datos remota, transformándolos con una herramienta ETL y cargándolos en un almacén de datos local. Utiliza una herramienta de análisis de negocios para presentar la información, creando un cuadro de mando principal y cinco informes sobre categorías específicas, como criticidad, país, estado, reabiertas, parches y causa raíz. Los resultados cumplieron las expectativas, logrando aspectos técnicos y de análisis de negocios al generar un almacén de datos y visualizar informes y gráficos sobre el comportamiento de las incidencias. (Hernandez, 2023)

Por otro lado, en el proyecto de implementación de inteligencia de negocios que determina el comportamiento de los estudiantes virtuales en el lms canvas, aplican la metodología crisp-dm que significa Cross-Industry Standard Process for Data Mining, la cual consiste en un método para guiar trabajos de minería de datos, esta metodología comprende de varias fases que son la comprensión del negocio, comprensión de los datos, preparación de datos, modelamiento, evaluación e implementación. Esta metodología les permitió entender las necesidades puntuales que tenía la organización, identificar y explorar las diferentes fuentes de datos internas, realizar limpiezas de datos, construir el modelo físico del repositorio de datos y realizar ingesta de datos. Posteriormente se desarrollaron reportes en la herramienta de inteligencia de negocios Power BI, permitiendo a los administradores del sistema de gestión de aprendizaje de la organización, visualizar y medir cómo los estudiantes están

llevando a cabo el desarrollo de las asignaciones dentro del curso, cuales son los módulos donde los estudiantes tienen más interacción, el % de avance del curso, adicionalmente identificar aquellos estudiantes que no realizan ninguna asignación o actividades en el curso permitiendo buscar estrategias pedagógicas que eviten retrasos en entregas de actividades. (Cáceres, 2020)

De los diversos proyectos encontrados sobre el comportamiento de los usuarios en aplicativos webs, se identifica que estos son una pauta y guía importante a tener durante el desarrollo de este proyecto, sin embargo, el desarrollo de los dashboard o reportes, no se relaciona específicamente con el comportamiento de los usuarios dentro de un aplicativo web referente a consulta de información de comercio exterior.

Acerca de los aplicativos de consulta de información de comercio exterior disponibles en el mercado como Legiscomex o CVN, no se posee información que permita identificar si las empresas proveedoras de estos tienen en cuenta el registro o las actividades de uso que tienen los usuarios con la plataforma de consulta. (Legiscomex, 2023), (CVN, 2023).

Con este proyecto se pretende tener en cuenta el registro de actividades de consulta de los usuarios para que el área comercial de SICEX la use posteriormente para proyectar su actividad de asesoramiento comercial hacia los usuarios de acuerdo con la búsqueda sistemática realizada por ellos.

3.3 Impacto

Con el resultado del Dashboard permitirá fortalecer el asesoramiento comercial sobre las consultas que requieren los usuarios que usan el aplicativo SICEX, facilitándole al área comercial identificar oportunidades de servicio y productos a la compañía con la actividad de los usuarios.

3.4 Componente de Innovación

En SICEX, la innovación se dará a partir de la explotación de la información semiestructurada que se guarda en la base de datos y que a la fecha no está siendo usada. Este proyecto permitirá generar nuevo conocimiento al área comercial a partir del Dashboard entregado, utilizando la información de manera más efectiva en la toma de decisiones en la creación y ofrecimiento de servicios.

Lo anterior, se traduce en beneficios económicos para la compañía, ya que permite un mejor asesoramiento en las necesidades de los usuarios del aplicativo.

4. Metodología

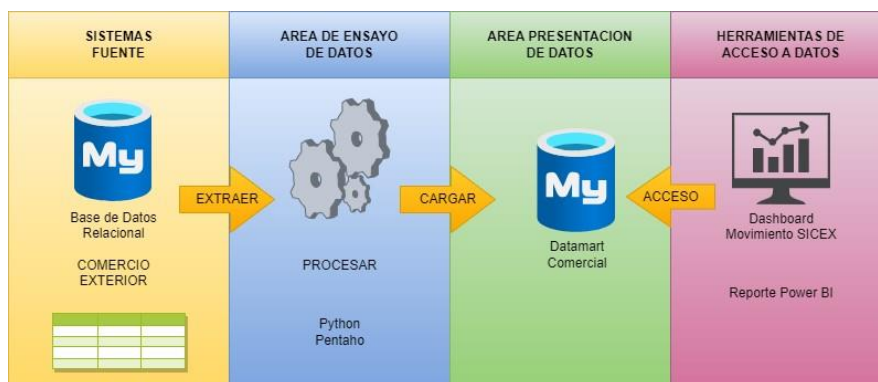
La metodología es la guía para el desarrollo de un proyecto y teniendo en cuenta la guía del conocimiento para la gestión de datos DAMA-DMBOK (segunda edición-2017) para el desarrollo de un Datawarehouse se pueden señalar dos de las más usadas, la metodología Inmon y la metodología Kimball, las cuales tienen diferentes enfoques para modelar y desarrollar el Datawarehouse. (DAMA, 2023)

Inmon define un Data Warehouse (almacén de datos) es un conjunto de datos organizados en torno a un tema específico, donde la información se encuentra integrada, se mantiene a lo largo del tiempo sin cambios y se utiliza para respaldar el proceso de toma de decisiones de la alta dirección. Para gestionar eficazmente estos datos, se emplea un modelo relacional normalizado. Ralph Kimball, por su parte, define un Data Warehouse como una réplica de los datos de transacciones, estructurada de manera específica para consultas y análisis. (International, 2017).

En resumen, la elección de la metodología para desarrollar el Dashboard de visualización de indicadores de uso de los clientes en el aplicativo web SICEX se basa en los requisitos específicos de la compañía y sus necesidades. En este caso, la metodología seleccionada es la de Kimball, que mejor se adapta a los objetivos de este proyecto. (Astera, 2023)

A continuación, se describen cada uno de los componentes de la arquitectura de un Datawarehouse según Kimball (ver Figura 1) (International, 2017):

Figura 1 Arquitectura Data Warehouse Kimball



Fuente: Elaboración propia a partir del uso de imágenes de Drawio.

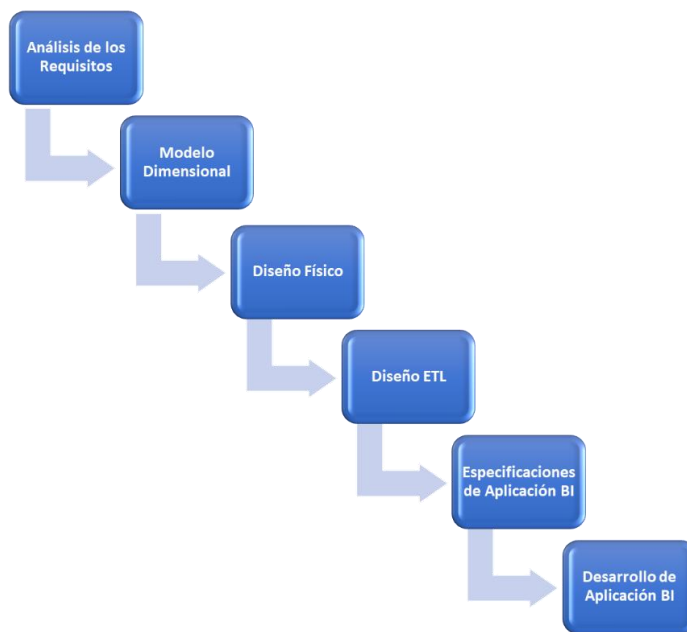
- **Sistemas Fuente:** Son las aplicaciones operacionales/transaccionales de la empresa.
- **Área de ensayo de datos:** El área de ensayo de datos de Kimball incluye el conjunto de procesos necesarios para integrar y transformar los datos para su presentación. En este caso se utiliza procesos ETL que se van a aplicar.
- **Área de presentación de datos:** Corresponde a los Datamarts un tipo de almacenamiento de datos utilizado para soportar capas de presentación de datos en el ambiente DW. Y en este proyecto el Datamart está orientado al monitoreo de la aplicación.
- **Herramientas de acceso a los datos:** Se centra en los requerimientos de datos de los usuarios finales.

La metodología Kimball se centra en la entrega de información útil y práctica a los usuarios finales, así como el modelo dimensional que es eficaz para consultas analíticas y su enfoque iterativo, permitiendo una entrega rápida de soluciones de inteligencia de negocios. (Astera, 2023)

DAMA reconoce en su capítulo Data Warehousing e inteligencia de negocios, la eficacia de esta metodología en proporcionar soluciones que agregan valor a las compañías a través de la gestión de datos, estructuración de datos de manera que sean fáciles de comprender y analizar, otorgando información valiosa de manera oportuna. (DAMA, 2023)

La siguiente figura describe el ciclo de desarrollo de la metodología Kimball con el cual se establece el desarrollo del proyecto (ver figura 2). (Gonzalez, 2021)

Figura 2 Metodología Kimball para el desarrollo del proyecto



Fuente: Elaboración Propia

- **Análisis de los requisitos:** Se parte de la necesidad del negocio, para desarrollar la solución que dé respuesta a dicha necesidad, identificando el valor que se le dará al usuario final.
- **Modelo dimensional:** Es el desarrollo del modelo lógico partiendo del entendimiento de la necesidad del negocio, donde se establece el nivel de granularidad de lo que se quiere trabajar, se identifican las dimensiones a partir de que sus atributos sean posibles candidatos para ser encabezado en los informes, tablas pivot o cualquier forma de visualización.

Además de identificar las medidas, las cuales son los atributos que se desea analizar, sumando o agrupando sus datos. Estas medidas se encuentran en las tablas de hechos o Fact, cada tabla de hechos tiene como atributos una o más medidas del proceso de negocio de acuerdo con el análisis de requisitos.

- **Diseño físico:** La fase de diseño físico se concentra en la elección de las estructuras esenciales para respaldar el modelo lógico, considerando factores como el tipo de base de datos, la eficiencia en el procesamiento de consultas y la gestión del almacén de datos.
- **Diseño ETL:** Esta actividad establece las operaciones que permiten extraer los datos, transformarlos y cargarlos en este caso al Datamart. En el proceso de extracción se obtienen los datos que permiten efectuar la carga al modelo físico definido. El proceso de transformación convierte los datos fuente para cargar el modelo físico. Y el proceso de carga de datos que pobla el Datamart.
- **Especificaciones de aplicación BI:** Esta fase es fundamental para definir los requisitos y la funcionalidad del sistema de BI, que incluye la creación del Dashboard para visualizar y analizar los datos.
- **Desarrollo de aplicación BI:** En esta etapa se desarrolla el tablero de control (o Dashboard) para el usuario final. En el tablero se encuentra la información relevante y específica para apoyar la toma de decisiones. (Pérez, 2018)

5. Desarrollo de la propuesta

A continuación, se presenta el desarrollo del proyecto para elaborar un Dashboard que tiene como objetivo la visualización de indicadores sobre el uso que tienen los clientes del aplicativo web SICEX. Sobre la base de la metodología planteada en el apartado anterior se describen las fases realizadas para el desarrollo del proyecto.

5.1 Análisis de los requisitos

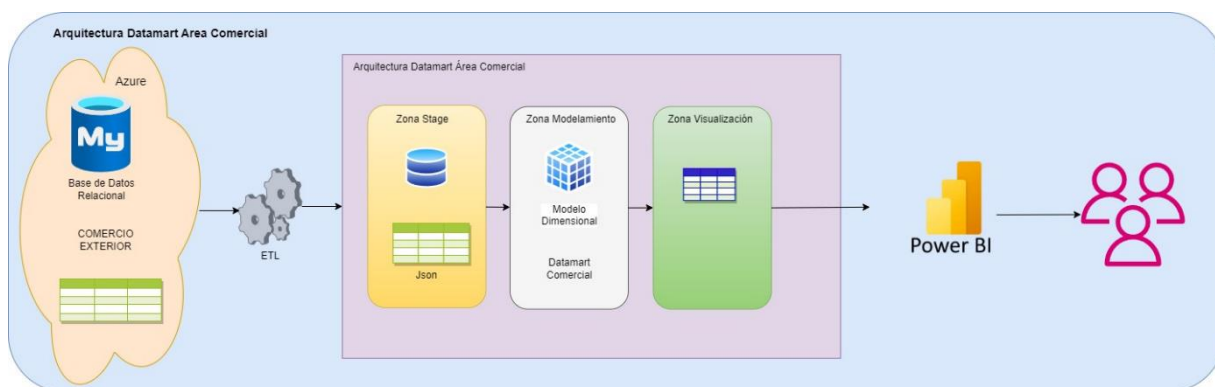
En esta fase se realiza el análisis de la base de datos MySQL llamada `sicex_r` en la cual se almacena la información relacionada con el aplicativo SICEX. En ella se identifica que en la tabla `monitoreo`, existe un campo `monitoreo_request`, el cual es un log que contiene información no estructurada de todas las actividades de consulta que realiza el cliente en el aplicativo web SICEX.

Este log contiene información relevante como el código arancelario que seleccionó, el país de consulta, el intercambio (Exportación o Importación) y los demás filtros que el cliente utilizó para la consulta. Adicionalmente, en la misma tabla se encuentran campos que se relacionan con el log, como el tipo de información de consulta (Aduanas, agentes marítimos, solicitudes de registro de importación), la acción (Dashboard o Información Detallada), el usuario que se logueó y la fecha en que realizó la acción.

Se requirió utilizar estos datos para comprender el uso de la aplicación por parte de los clientes y a través de la visualización permitir al área comercial conocer las preferencias y comportamientos de los clientes en el uso de la información.

Por lo anterior, se diseña el proceso a realizar el cual se representa en la figura siguiente (Ver Figura 3).

Figura 3 Arquitectura Datamart



Fuente: Elaboración propia a partir del uso de imágenes de Drawio

5.2 Modelo Dimensional

A partir del análisis de todas las tablas de la base de datos `sicex_r`, en esta fase se definen cuales tablas específicamente se trabajaron, cuales atributos son los necesarios para suplir la necesidad para el área comercial. Se identifican las dimensiones necesarias a partir de los atributos que son candidatos para los filtros de los informes y se establece el nivel de granularidad.

Las Dimensiones identificadas para el proyecto son:

- `dimCalendario`
- `dimAccion`
- `dimProducto`
- `dimIntercambio`
- `dimPais`
- `dimEmpresa`
- `dimUsuario`
- `dimCapitulo`
- `dimPartida`
- `DimSubpartida`

Las tablas de dimensiones mencionadas anteriormente definen la granularidad que permiten un análisis sobre las tablas de hechos.

Se definen dos tablas de hechos, una que almacena la cantidad de veces que se loguea un usuario en una fecha determinada y la otra almacena las actividades que realiza el usuario dentro el aplicativo.

Las tablas de hechos son:

- `monitoreoAplicacion`
- `sesionAplicacion`

Cada una de las tablas de dimensiones y de hechos anteriormente mencionadas, cumple un papel específico en el proyecto y para el análisis de datos. A continuación, se relaciona la descripción de cada una de ellas.

- **dimCalendario (Dimensión Calendario):** Contiene información relacionada con el calendario, como fechas, días de la semana, meses, trimestres y años que identifica el uso del aplicativo.
- **dimAccion (Dimensión Acción):** Registra las acciones o actividades que los usuarios realizan en el aplicativo, como consulta del Dashboard, información detallada y descarga de archivos de con información detallada en formato Excel.

- **dimProducto (Dimensión Producto):** Registra los diferentes productos que se encuentran habilitados en el aplicativo, ejemplo SISDUAN, SISMAR, Registros de importación.
- **dimIntercambio (Dimensión Intercambio):** Registra el tipo de consulta que realiza sobre el producto, si es una exportación o una importación.
- **dimPais (Dimensión País):** Registra los diferentes países habilitados para consulta en el aplicativo.
- **dimEmpresa (Dimensión Empresa):** Registra las empresas (Clientes) que se encuentran habilitados para el acceso al aplicativo.
- **dimUsuario (Dimensión Usuario):** Registra los usuarios que se encuentran asociados a una empresa y tiene autorización para el uso del aplicativo.
- **DimCapitulo (Dimensión Capitulo):** Registra la clasificación arancelaria de las mercancías consultadas en el aplicativo, donde la estructura del código es de 2 dígitos identificando de forma general las mercancías.
- **DimPartida (Dimensión Partida):** Registra la clasificación arancelaria de las mercancías consultadas en el aplicativo, donde la estructura del código es de 4 dígitos identificando de forma más detallada las mercancías, respecto a la dimensión de Capitulo.
- **dimSubpartida (Dimensión Subpartida):** Registra la clasificación arancelaria de las mercancías consultadas en el aplicativo, donde la estructura del código es de 6 dígitos identificando de forma más detallada las mercancías, respecto a la dimensión de Capitulo y dimensión Partida.
- **monitoreoAplicación (Tabla de Hechos):** Registra eventos específicos relacionados con el uso del aplicativo web, las interacciones de los usuarios, como el número de consultas, descargas, filtros aplicados y cantidad de registros consultados. Esta tabla de hechos contiene información que permite realizar análisis y generar informes sobre el uso del aplicativo, así como para evaluar el comportamiento de los usuarios.
- **SesionAplicacion (Tabla de Hechos):** Registra la cantidad de veces en que los usuarios realizan Loguin al aplicativo. Se podrá realizar un análisis de los usuarios que se encuentran activos, para validar cuales de ellos son los que más utilizan el aplicativo y quienes son los que no la utilizan.

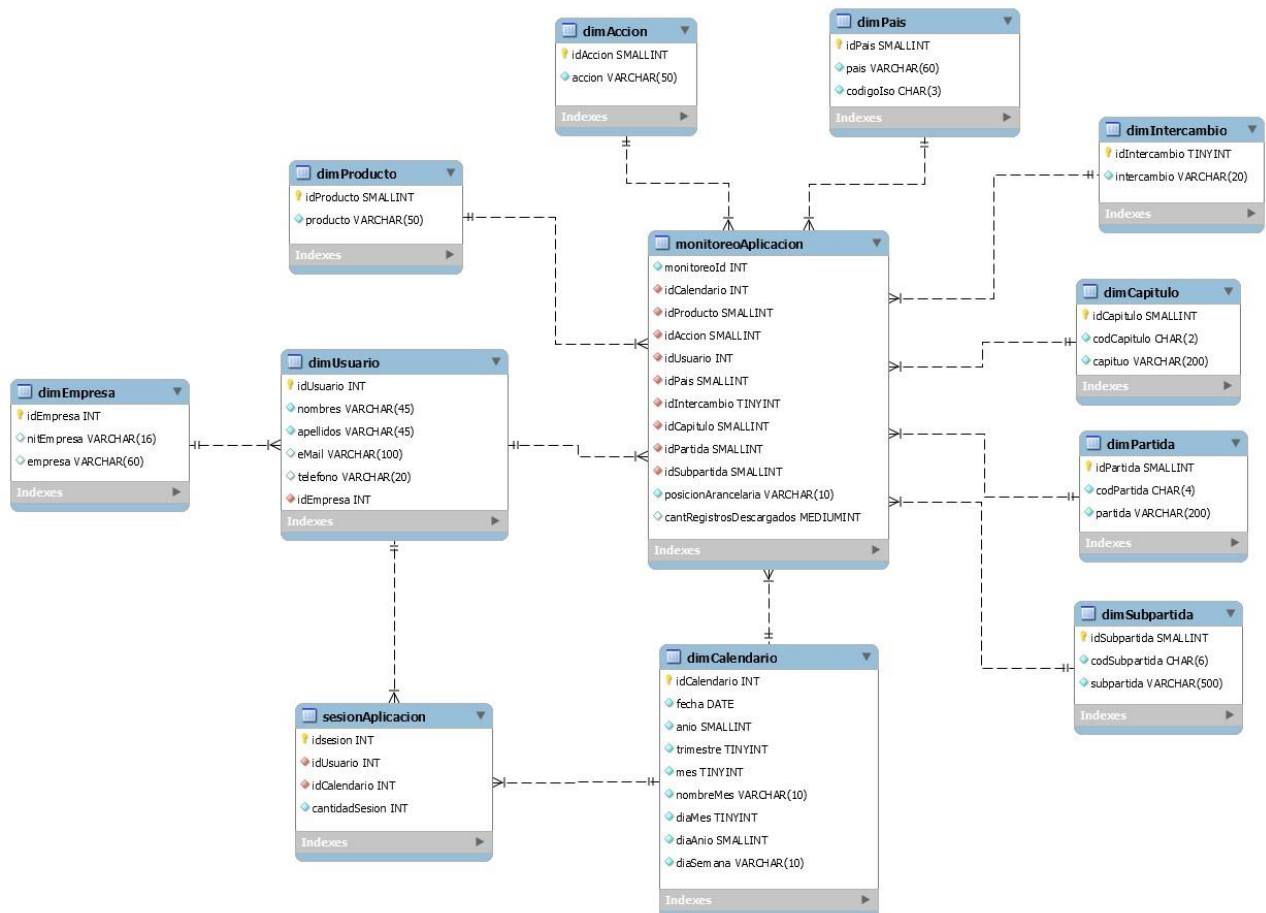
5.3 Diseño físico

5.3.1 Diseño del Modelo Físico

De acuerdo con las tablas de dimensiones y tablas de hechos identificadas en la fase anterior, se utilizó MySQL Workbench para construir el diseño del modelo físico para la bodega de datos, se utilizó la tipología tipo constelación que se basa en varias tablas de dimensiones y de hechos que permite almacenar toda la información necesaria para responder a las necesidades del proyecto.

Los tipos de datos definidos para cada tabla se establecieron considerando a la base de datos fuente sicex_r. Se establecieron claves primarias y foráneas para mantener la integridad de los datos y permitir relaciones entre tablas de dimensiones y de hechos (Ver Figura 4).

Figura 4 Modelo Físico Bodega de Datos



Fuente: Elaboración Propia

5.3.2 Implementación del modelo Físico

De acuerdo con el modelo físico construido se creó el script para la implementación en un servidor de base de datos MySQL. Ver anexo Script DDLBodegadedatos.

5.3.3 Diccionario de Datos:

Se relaciona la documentación del diccionario de datos el cual describe cada una de las tablas que tiene la bodega de datos como: El nombre de los campos, su descripción, tipos de datos, relaciones, longitudes y restricciones. Ver anexo Diccionario de Datos.

5.4 Diseño ETL

En esta fase se relaciona y prepara la información necesaria para alimentar las dimensiones y las tablas de hechos de la bodega de datos. A continuación, se describe el proceso ETL construido.

Se desarrollan tres scripts utilizando el lenguaje de programación Python, los cuales tienen como objetivo realizar todo el procesamiento de datos desde el origen hasta la bodega de datos.

El primer script llamado ConexionBD, contiene la parametrización de la conexión a los servidores `mysql_cloud_sicex_r`, `mysql_reporting` y `mysql_dw_oper`. (Ver anexo 3).

El segundo script llamado ConexionCorreo.py es la configuración y la parametrización para enviar el correo de confirmación exitosa o no, de la ejecución del proceso ETL (Ver anexo 4).

El tercer y último script llamado ETL.py corresponde al script principal del proceso ETL, el cual permite realizar la extracción de datos de la fuente `sicex_r` ubicada en la nube, apoyándose en el primer script ConexionBD. (Ver anexo 5).

El código en Python se encarga de extraer datos de la base de datos en la nube y de bases de datos locales, realiza transformaciones en esos datos y cargar en la bodega de datos. Cada sección del código se encarga de actualizar las dimensiones y hechos en la bodega de datos a partir de los datos obtenidos.

El proceso ETL se describe a continuación:

Se realiza la conexión al servidor en la nube `mysql_cloud_sicex_r` y `mysql_dw_oper`, el cual tiene la data fuente, posteriormente al servidor `mysql_reporting` donde se encuentra alojada la bodega de datos. (Ver anexo 5).

De la conexión `mysql_cloud_sicex_r`, se extraen datos de las siguientes tablas, relevantes para la ingesta en la bodega de datos (Ver anexo 5).

- monitoreo
- accion
- producto
- intercambio
- pais
- empresa
- usuario
- session_tokens

Y se cargan en la tabla de dimensiones correspondientes (Ver anexo 5).

De la conexión mysql_dw_oper, se extraen datos de las siguientes tablas, relevantes para la ingesta en la bodega de datos (Ver anexo 5).

- ✓ capitulo
- ✓ partida
- ✓ subpartida

Y se cargan en la tabla de dimensiones correspondientes.

De las tablas mencionadas anteriormente se utilizan algunos campos que garantizan la granularidad en la bodega de datos. Posteriormente, se realiza la extracción y carga hacia las tablas de hechos sesionAplicacion y monitoreoAplicacion. Dentro del campo **monitoreo_request** (Ver anexo 5).

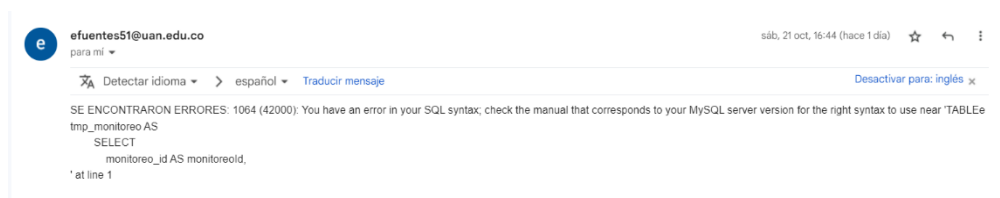
5.4.1 Pruebas de ejecución del proceso ETL

En el caso que la ejecución del proceso ETL sea exitosa, se llama al script ConexionCorreo.py para enviar un correo electrónico indicando que el proceso fue ejecutado exitosamente. (Ver anexo 5).

En el caso que genere un error en el proceso, se envía un correo con los detalles del error (Ver anexo 5).

Se relaciona el correo recibido en caso de generarse un error en el proceso (Ver Figura 6).

Figura 5 Evidencia Envío Correo Cuando el Proceso ETL no es Exitoso



Fuente: Elaboración Propia

Se relaciona el correo recibido de que la ejecución finalizó exitosamente (Ver Figura 7).

Figura 6 Evidencia Envío Correo Cuando el Proceso ETL es Exitoso



Fuente: Elaboración Propia

Al terminar el proceso ETL, se muestra el mensaje de ejecución exitosa (Ver figura 8).

Figura 7 Mensaje Confirmación Proceso ETL Exitoso

```
[Running] python -u "c:\Users\lfuentes\Documents\BACKUP\Personal\UAN\proyectoGrado\ETL_Python\ETL.py"
Correo enviado con éxito.
Correo enviado con éxito.
FINALIZADO
[Done] exited with code=0 in 367.275 seconds
```

Fuente: Elaboración Propia

En caso de no ser exitosa la ejecución del proceso ETL, se genera un mensaje indicando el problema o error presentado (Ver Figura 9).

Figura 8 Mensaje Confirmación Proceso ETL no Exitoso

```
[Running] python -u "c:\Users\lfuentes\Documents\BACKUP\Personal\UAN\proyectoGrado\ETL_Python\ETL.py"
Correo enviado con éxito.
SE ENCONTRARON ERRORES: 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'TABLEE
tmp_monitoreo AS
SELECT
  monitoreo_id AS monitoreoId,
' at line 1
Correo enviado con éxito.
FINALIZADO
[Done] exited with code=0 in 5.212 seconds
```

Fuente: Elaboración Propia

5.4.2 Validación y Pruebas Cargue a la bodega de Datos

Luego de la ejecución del proceso ETL, se evidencia la ingesta de datos a cada una de las tablas dimensiones y hechos que actualmente tiene la bodega de datos, a continuación, se muestra una evidencia de información contenida en cada una de las tablas (Ver Figuras desde la 10 hasta la 21).

Dimensión dimCalendario:

Figura 9 Validación Data Tabla dimCalendario

idCalendario	fecha	anio	trimestre	mes	nombreMes	diaMes	diaAnio	diaSemana
448	2023-09-01	2023	3	9	SEPTIEMBRE	1	244	VIERNES
449	2023-09-02	2023	3	9	SEPTIEMBRE	2	245	SÁBADO
450	2023-09-03	2023	3	9	SEPTIEMBRE	3	246	DOMINGO
451	2023-09-04	2023	3	9	SEPTIEMBRE	4	247	LUNES
452	2023-09-05	2023	3	9	SEPTIEMBRE	5	248	MARTES
453	2023-09-06	2023	3	9	SEPTIEMBRE	6	249	MIÉRCOLES
454	2023-09-07	2023	3	9	SEPTIEMBRE	7	250	JUEVES
455	2023-09-08	2023	3	9	SEPTIEMBRE	8	251	VIERNES
456	2023-09-09	2023	3	9	SEPTIEMBRE	9	252	SÁBADO
457	2023-09-10	2023	3	9	SEPTIEMBRE	10	253	DOMINGO
458	2023-09-11	2023	3	9	SEPTIEMBRE	11	254	LUNES
459	2023-09-12	2023	3	9	SEPTIEMBRE	12	255	MARTES

Fuente: Elaboración Propia

Dimensión dimAccion:

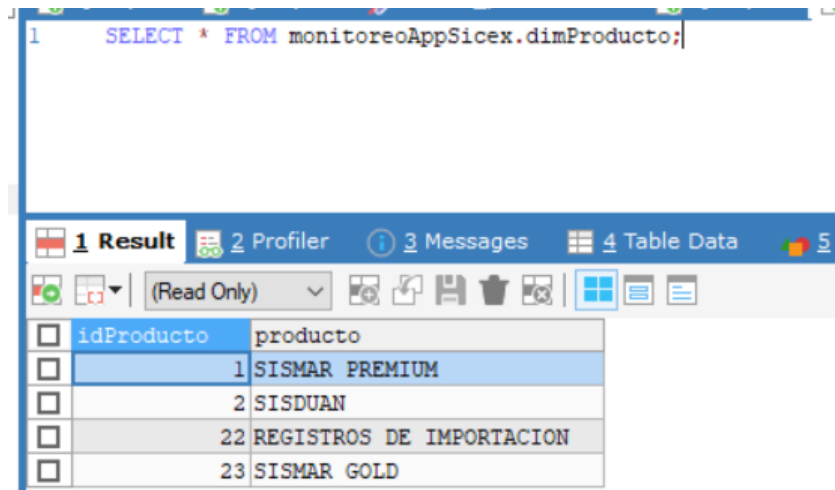
Figura 10 Validación Data Tabla dimAccion

idAccion	accion
400	Dashboard
410	Dashboard - PDF
500	Pivot
600	Pivot - Excel
700	Detailed
800	Detailed - Excel

Fuente: Elaboración Propia

Dimensión dimProducto:

Figura 11 Validación Data Tabla dimProducto



The screenshot shows a SQL query window with the following text:

```
1 SELECT * FROM monitoreoAppSicex.dimProducto;
```

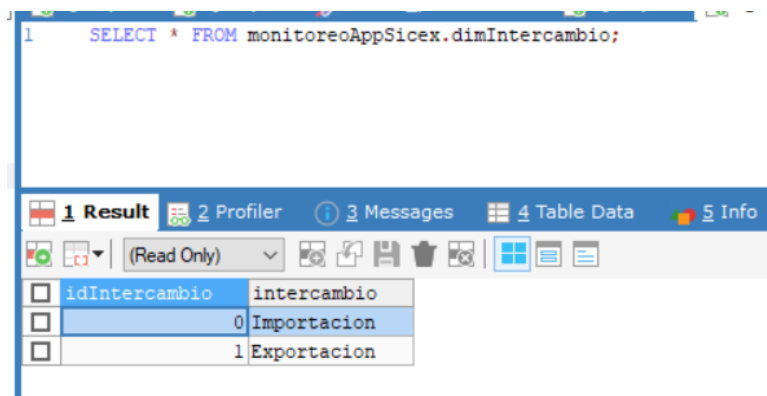
The result set is displayed in a table with the following columns and rows:

idProducto	producto
1	SISMAR PREMIUM
2	SISDUAN
22	REGISTROS DE IMPORTACION
23	SISMAR GOLD

Fuente: Elaboración Propia

Dimensión dimIntercambio:

Figura 12 Validación Data Tabla dimIntercambio



The screenshot shows a SQL query window with the following text:

```
1 SELECT * FROM monitoreoAppSicex.dimIntercambio;
```

The result set is displayed in a table with the following columns and rows:

idIntercambio	intercambio
0	Importacion
1	Exportacion

Fuente: Elaboración Propia

Dimensión dimPais:

Figura 13 Validación Data Tabla dimPais

idPais	pais	codigoIso
1	Colombia	CO
2	Ecuador	EC
3	Peru	PE
4	Chile	CL
6	Uruguay	UY
7	Paraguay	PY
11	Espana	ES
100	Estados Unidos	US
101	Brasil	BR
102	Panama	PA
103	Venezuela	VE
105	Mexico	MX
106	Inglaterra Non EU	GB
107	El Salvador	SV
108	India	IN
109	Indonesia	ID
110	Thailandia	TH
111	Korea	KR

Fuente: Elaboración Propia

Dimensión dimEmpresa:

Figura 14 Validación Data Tabla dimEmpresa

idEmpresa	nitEmpresa	empresa
28892	900176224	C.I SAN JORGE 714 SAS
71107	800011987	TENARIS TUBOCARIBE LTDA.
80128	800021013	CENTRO ACEROS S A
83191	800024075	COLTRANS SA
85947	800026845	OCATI LTDA
93900	800034825	C I ANTILLANA SA
99057	800039996	KUEHNE NAGEL S. A. S.
100064	800041007	PROVIMI S.A. ANTES BIOVET S.A.
103850	800044807	STEPAN COLOMBIA S.A.S.
106063	800047031	DURATEX SA
118282	800059470	POLIPROPILENO DEL CARIBE S.A.
125928	800067129	LEH COLOMBIA SAS
129111	800070312	SILICAR LTDA
136629	800077828	FACTORES Y MERCADEO S A
149388	800090584	EMBAJADA ARGENTINA EN COLOMBIA
149660	800090786	EMBAJADA DE BRASIL - SECTOR COMERCIAL

Fuente: Elaboración Propia

Dimensión dimUsuario:

Figura 15 Validación Data Tabla dimUsuario

The screenshot shows a SQL query result in a database management tool. The query is: `SELECT * FROM monitoreoAppSicex.dimUsuario;` The result set contains 10 rows of user data with columns: idUsuario, nombres, apellidos, eMail, and telefono.

idUsuario	nombres	apellidos	eMail	telefono
5	JAIIME QR	CASTANO	jcastano@sicex.com	3693390
9	LIBARDO	GALLO	lgallo@sicex.com.co	3693390
11	FABLO	CASTANO	pcastano@sicex.com.co	3693390
30	JOHAN	FINEROS	analistacomercial@coltrans.com.co	4239300
31	JOHAN	FINEROS	analistacomercial@coltrans.com.co	4239300
32	MILLY	MANCHEGO PEREZ	administracion@andesia.com.co	5467000 ext. 104
33	NATHALEE	MONTAÑEZ	nathalee.montanez@epiroc.com	4199200
34	DIEGO A	VEGA	diego.vega@co.atlascopco.com	4199200
42	MARIA PAULA	GARCIA	maria-paula.garcia@basf.com	6322260
44	FABIAN	ESCOBAR HENAO	ASISTENTE2.IMPO@camejia.com	4446767

Fuente: Elaboración Propia

Dimensión dimCapitulo:

Figura 16 Validación Data Tabla dimCapitulo

The screenshot shows a SQL query result in a database management tool. The query is: `SELECT * FROM monitoreoAppSicex.dimCapitulo;` The result set contains 10 rows of chapter data with columns: idCapitulo, codCapitulo, and capitulo.

idCapitulo	codCapitulo	capitulo
2201	00	DESCONOCIDO
2202	01	ANIMALES VIVOS
2203	02	CARNE Y DESPOJOS COMESTIBLES.
2204	03	PESCADOS Y CRUSTACEOS, MOLUSCOS Y DMS INVERTEBRADOS ACUATICOS.
2205	04	LECHE Y PRODUCTOS LACTEOS, HUEVOS DE AVE, MIEL NATURAL, PRODUCTOS C
2206	05	LOS DMS PRODUCTOS DE ORIGEN ANIMAL NO EXPRESADOS NI COMPREDIDOS EN
2207	06	PLANTAS VIVAS Y PRODUCTOS DE LA FLORICULTURA.
2208	07	HORTALIZAS, PLANTAS, RAICES Y TUBERCULOS ALIMENTICIOS.
2209	08	FRUTAS Y FRUTOS COMESTIBLES, CORTEZAS DE AGRIOS (CITRICOS), MELONES
2210	09	CAFE TE YERBA MATE Y ESPECIAS.

Fuente: Elaboración Propia

Dimensión dimPartida:

Figura 17 Validación Data Tabla dimPartida

idPartida	codPartida	partida
26881	0101	CABALLOS- ASNOS- MULOS Y BURDEGANOS- VIVOS
26882	0102	ANIMALES VIVOS DE LA ESPECIE BOVINA
26883	0103	ANIMALES VIVOS DE LA ESPECIE PORCINA
26884	0104	ANIMALES VIVOS DE LAS ESPECIES OVINA O CAPRINA
26885	0105	GALLOS- GALLINAS- PATOS- GANSOS- PAVOS (GALLIPAVOS) Y PINTADAS- DE LAS ESPECIES DOM
26886	0106	LOS DEMAS ANIMALES VIVOS
26887	0201	CARNE DE ANIMALES DE LA ESPECIE BOVINA- FRESCA O REFRIGERADA
26888	0202	CARNE DE ANIMALES DE LA ESPECIE BOVINA- CONGELADA
26889	0203	CARNE DE ANIMALES DE LA ESPECIE PORCINA- FRESCA- REFRIGERADA O CONGELADA
26890	0204	CARNE DE ANIMALES DE LAS ESPECIES OVINA O CAPRINA- FRESCA- REFRIGERADA O CONGELADA
26891	0205	CARNE DE ANIMALES DE LAS ESPECIES CABALLAR ASNAL O MULAR FRESCA REFRIGERADA O CONGE
26892	0206	DESPOJOS COMESTIBLES DE ANIMALES DE LAS ESPECIES BOVINA- PORCINA- OVINA- CAPRINA- C
26893	0207	CARNE Y DESPOJOS COMESTIBLES- DE AVES DE LA PARTIDA 01.05- FRESCOS- REFRIGERADOS O
26894	0208	LAS DEMAS CARNES Y DESPOJOS COMESTIBLES- FRESCOS- REFRIGERADOS O CONGELADOS
26895	0209	TACTINO SIN PARTES MEGRAS Y GRASA DE CERDO O DE AVE SIN FUNDIR NI EXTRAER DE OTRO M

Fuente: Elaboración Propia

Dimensión dimSubpartida:

Figura 18 Validación Data Tabla dimSubpartida

idSubpartida	codSubpartida	subpartida
65535	99YYYY	DISPOSICIONES DE TRATO ESPECIAL

Fuente: Elaboración Propia

Tabla de hechos monitoreoAplicacion:

Figura 19 Validación Data Tabla monitoreoAplicacion

monitoreoId	idCalendario	idProducto	idAccion	idUsuario	idPais
10978667	448	2	700	7544	112
10978668	448	2	800	7544	112
10978669	448	2	700	7544	112
10978670	448	2	800	7544	112
10978671	448	2	700	7544	112
10978672	448	2	700	7544	112
10978673	448	2	800	7544	112
10978674	448	2	400	352	2
10978675	448	2	700	7544	112
10978676	448	2	800	7544	112
10978677	448	2	700	7544	112
10978678	448	2	400	352	2

Fuente: Elaboración Propia

Tabla de hechos sesionAplicacion:

Figura 20 Validación Data Tabla sesionAplicacion

idsesion	idUsuario	idCalendario	cantidadSesiones
42431	352	448	2
42432	7321	448	1
42433	8416	448	2
42434	3363	448	1
42435	8231	448	1
42436	9957	448	4
42437	10448	448	1
42438	9998	448	3
42439	6205	448	2
42440	10381	448	1
42441	10063	448	1
42442	10179	448	1
42443	5621	448	1
42444	9292	448	1
42445	1777	448	1
42446	2178	448	5
42447	8095	448	3
42448	9276	448	2

Fuente: Elaboración Propia

5.5 Especificaciones de aplicación BI

Esta fase es fundamental en la definición los requisitos y la funcionalidad del sistema de BI, que incluye la creación del Dashboard para visualizar y analizar los datos. Por lo tanto, se realizan las siguientes actividades:

5.5.1 Análisis de caso de uso

Para la información cargada en la bodega de datos y de acuerdo con las dimensiones y tablas de hechos, se identifican los siguientes escenarios para el uso del Dashboard.

5.5.2 Definición de métricas clave del Dashboard

En esta actividad se identifica las métricas clave a presentar en el Dashboard para ampliar el conocimiento de los asesores comerciales en cuanto a la actividad de los usuarios en el aplicativo web. En la siguiente tabla se muestra detalladamente cada una de las preguntas que se desea resolver, cuál es su descripción funcional, cuál cálculo se realiza para responder a dicha pregunta, y qué tipo de gráfico se utiliza para representar de forma fácil la respuesta hacia cada una de las preguntas que se encuentran en la tabla (Ver Tabla 1).

Tabla 1 Métricas clave del Dashboard

Pregunta	Descripción Funcional	Cálculo	Presentación con Power BI
Qué clientes y cuáles de sus usuarios se loguean o ingresan al aplicativo Web	Este indicador tiene como objetivo identificar quiénes son los usuarios que utilizan la aplicación y quiénes no. Proporciona una visión general de la adopción de la aplicación.	Calcula el número total de usuarios registrados en la aplicación y compararlo con el número de usuarios que han iniciado sesión o realizado alguna actividad en la aplicación. Se cuenta la cantidad de sesiones registradas en la tabla de sesiones de aplicación (Hechos Sesiones). Aquellos usuarios que tienen una o más sesiones se consideran como "utilizan el aplicativo", mientras que los que no tienen ninguna sesión se consideran como "no utilizan el aplicativo".	Se muestra en un gráfico de donut que destaque la proporción de usuarios que utilizan la aplicación y los que no.
Cuál es la participación de descarga de información por	Este indicador analiza quiénes son los clientes y usuarios que realizan descargas	Contabiliza el número de descargas realizadas por cada cliente y usuario, y calcula su participación en el total de descargas.	Utiliza gráficos de barras apiladas o tablas para mostrar la participación de descargas por

cliente y por usuario	en la aplicación y cuántas descargas realizan.	Se realiza un recuento de las acciones de descarga (o cualquier acción específica que represente una descarga) en la tabla de hechos Monitoreo Aplicación. Esto te dará la cantidad de descargas por cliente y usuario.	cliente y usuario. Se incorpora una tabla dinámica para permitir a los usuarios filtrar y explorar los datos.
Cuál es la participación de descarga de información por producto (SISMAR, SISDUAN, ETC)	Este indicador se centra en los productos específicos que se descargan en la aplicación y su popularidad.	Contabiliza el número de descargas por producto y calcula su participación en el total de descargas. Se realiza un recuento de las acciones de descarga (o la acción específica que representa una descarga) en la tabla de hechos Monitoreo Aplicación, agrupado por producto. Esto te dará la cantidad de descargas por producto.	Utiliza un gráfico de barras o una tabla o un gráfico de columnas agrupadas para mostrar la participación de descargas por producto. Se agregan etiquetas de datos para mostrar los valores numéricos.
Cuál es la participación de descarga de información por países del producto SISDUAN	Este indicador analiza cuales son los países de consulta que más comportamiento tienen respecto a descarga de información.	Contabiliza el número de descargas por país y calcula su participación en el total de descargas. Se filtra las acciones relacionadas con SISDUAN en la tabla de hechos Monitoreo Aplicación y agrupa los resultados por país. Esto te dará la cantidad de descargas por país relacionadas con SISDUAN.	Utiliza un mapa o un gráfico de barras para mostrar la participación de descargas por país. Los colores o las alturas de las barras indican la cantidad de descargas.
Cuál es la participación de las partidas arancelarias por capítulo, partida o subpartida que se consultan o se descarga información del aplicativo	Este indicador analiza cuales son las mercancías que más se consulta en el aplicativo.	Contabiliza el número de descargas o consultas de las mercancías y calcula su participación en el total. Se filtra las acciones relacionadas con las partidas arancelarias en la tabla de hechos Monitoreo Aplicación y agrupa los resultados por actividad. Esto te dará la cantidad de descargas o consultas por partidas arancelarias.	Utiliza tablas para mostrar la participación de descargas por partidas arancelarias. Se incorpora una tabla dinámica para permitir a los usuarios filtrar y explorar los datos

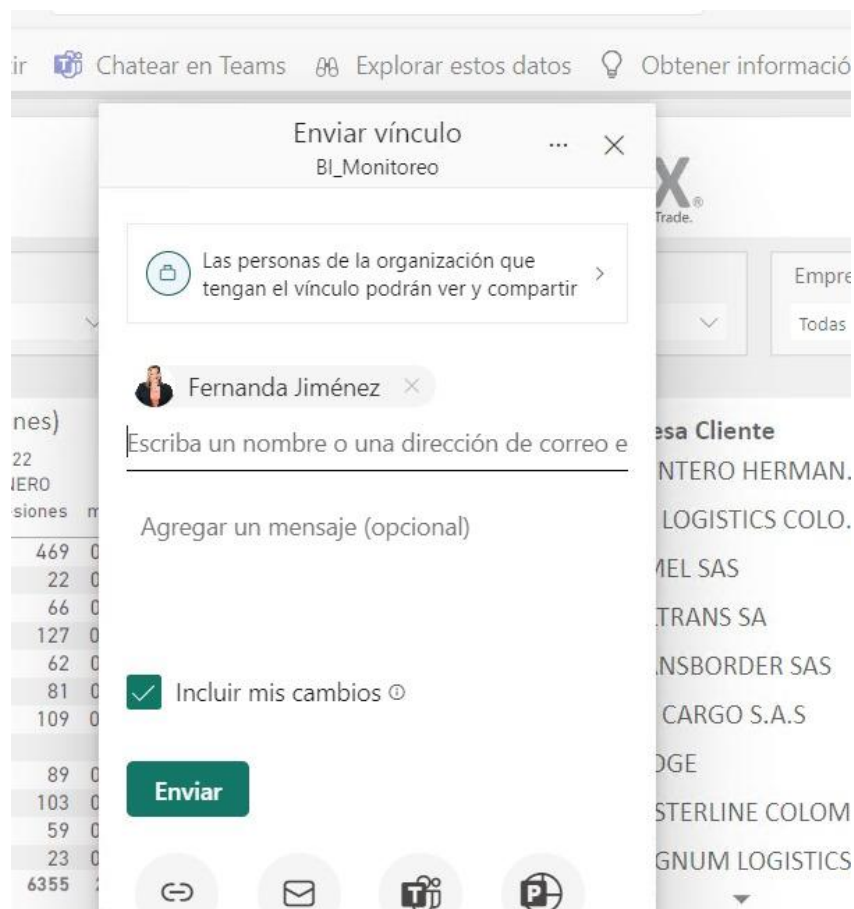
Fuente: Elaboración Propia

Al definir estas métricas clave, se crea un conjunto de indicadores que ayuda a los asesores del área comercial a comprender y evaluar la actividad en el aplicativo web SICEX de manera integral. Estas métricas proporcionan una visión completa de cómo los usuarios interactúan con la aplicación y permiten tomar decisiones basadas en datos para mejorar la experiencia del usuario. La periodicidad de actualización de los informes será mensualmente, durante la primera semana de cada mes.

5.5.3 Acceso al Dashboard y Clasificación de Usuarios

Para garantizar que los usuarios tengan acceso a la información relevante y que se mantenga la seguridad de los datos, el Dashboard se entregará a la coordinación del área comercial para que la definición de accesos sea identificada dentro de su equipo de trabajo. (Ver Figura 21)

Figura 21 Configuración de Acceso al Dashboard



Fuente: Elaboración Propia

5.6 Desarrollo de aplicación BI

En esta fase, se desarrolla e implementa la solución de BI a través de la herramienta Power BI, según lo establecido en etapas anteriores, más la creación de reportes, cuadros de mando, indicadores.

5.6.1 Conexión a Bodega de datos

Inicialmente se realiza los parámetros de conexión a la bodega de datos (Ver Figura 22).

Figura 22 Power BI Conexión a Bodega de datos

Base de datos MySQL

Servidor
172.16.1.34:3373

Base de datos
monitoreoAppSicex

▲ Opciones avanzadas

Tiempo de espera del comando en minutos (opcional)

Instrucción SQL (opcional, requiere una base de datos)

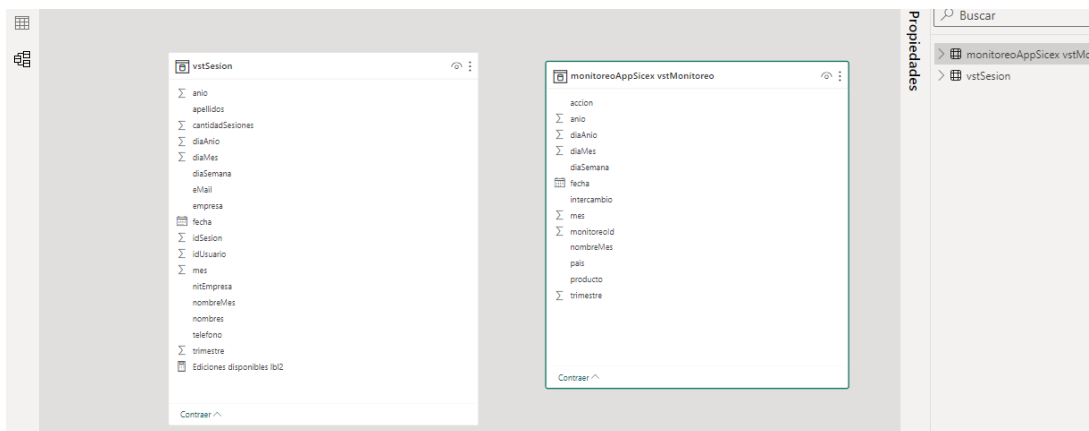
```
SELECT * FROM vstSesion
```

Fuente: Elaboración Propia

5.6.2 Selección Conjunto de Datos

Se establecen las tablas que son necesarias para la construcción de los indicadores (Ver Figura 23).

Figura 23 Power BI Estructura de Tablas



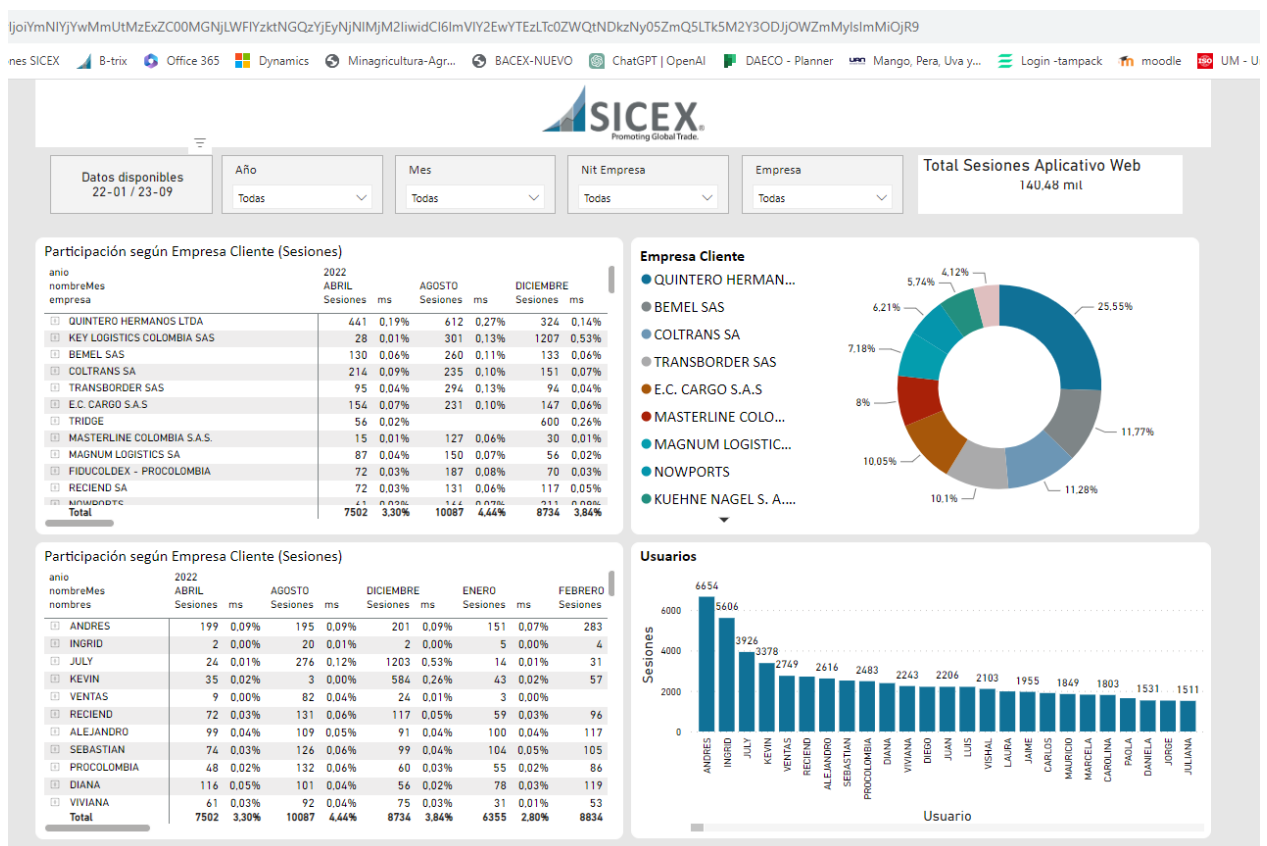
Fuente: Elaboración Propia

5.6.3 Desarrollo y visualización de los Dashboard

Se da inicio al desarrollo de cada uno de los tableros necesarios para responder a cada una de las preguntas que se establecieron en la fase de Especificaciones de Aplicaciones BI, donde se tienen en cuenta parámetros de visualización, colores y estilos de acuerdo con el manual de marca de la Compañía Quintero Hermanos.

Aquí se muestra de forma general, como queda construido el Dashboard con sus respectivas gráficos y tablas que permiten dar respuesta a cada una de las preguntas (Ver Figura 24).

Figura 24 Dashboard General



Fuente: Elaboración Propia

En la siguiente figura se muestra cómo se responde a alguna de las preguntas de negocio, como: Qué clientes y cuáles de sus usuarios se loguean o ingresan al aplicativo Web. Se podrá visualizar en una tabla de forma detallada cada uno de los clientes que ingresaron o se loguearon en la plataforma (Ver Figura 25).

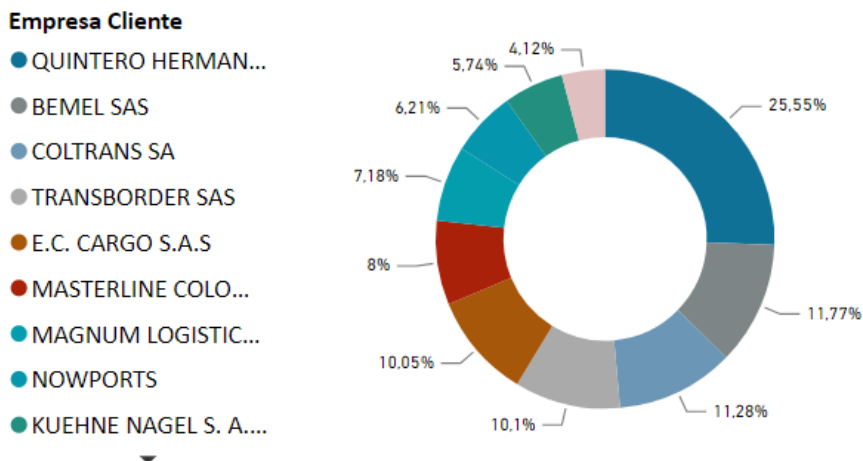
Figura 25 Power BI Participación Según Empresa Cliente

ano nombreMes empresa	2023 SEPTIEMBRE		Total		Total	
	Sesiones	ms	Sesiones	ms	Sesiones	ms
QUINTERO HERMANOS LTDA	544	0,24%	4662	2,05%	11099	4,88%
KEY LOGISTICS COLOMBIA SAS	141	0,06%	5606	2,47%	9636	4,24%
BEMEL SAS	284	0,12%	2928	1,29%	5113	2,25%
COLTRANS SA	325	0,14%	2572	1,13%	4899	2,16%
TRANSBORDER SAS	255	0,11%	2503	1,10%	4388	1,93%
E.C. CARGO S.A.S	172	0,08%	2089	0,92%	4368	1,92%
TRIDGE	13	0,01%	2585	1,14%	3802	1,67%
MASTERLINE COLOMBIA S.A.S.	435	0,19%	2862	1,26%	3478	1,53%
MAGNUM LOGISTICS SA	273	0,12%	1780	0,78%	3120	1,37%
FIDUCOLDEX - PROCOLOMBIA	181	0,08%	1420	0,62%	2960	1,30%
RECIEND SA	275	0,12%	1493	0,66%	2710	1,19%
NOWPORTS	22	0,01%	944	0,42%	2207	1,00%
Total	13939	6,13%	119125	52,42%	227264	100,00%

Fuente: Elaboración Propia

Incluso el Top 10 de las empresas Cliente que más han ingresado al aplicativo (Ver Figura 26).

Figura 26 Power BI Empresa Cliente Top 10



Fuente: Elaboración Propia

Participación de descarga de información por países del producto SISDUAN (Ver Figura 27 y 28).

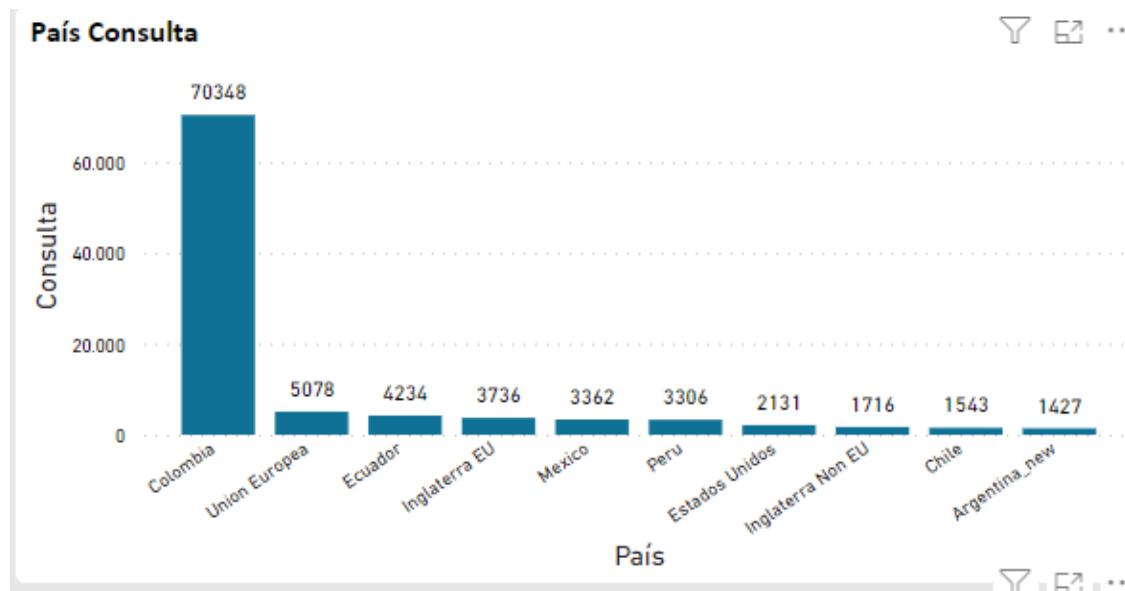
Figura 27 Power BI Participación Países

Participación según País Consulta ()

País Consulta	2023	Total
Colombia	70348	70348
Union Europea	5078	5078
Ecuador	4234	4234
Inglaterra EU	3736	3736
Mexico	3362	3362
Peru	3306	3306
Estados Unidos	2131	2131
Inglaterra Non EU	1716	1716
Chile	1543	1543
Argentina_new	1427	1427
Panama	895	895
Venezuela	885	885
Brasil	753	753
Costa Rica	696	696
Indonesia	516	516
Guatemala	500	500
Total	105361	105361

Fuente: Elaboración Propia

Figura 28 Power BI Top 10 Participación Países



Fuente: Elaboración Propia

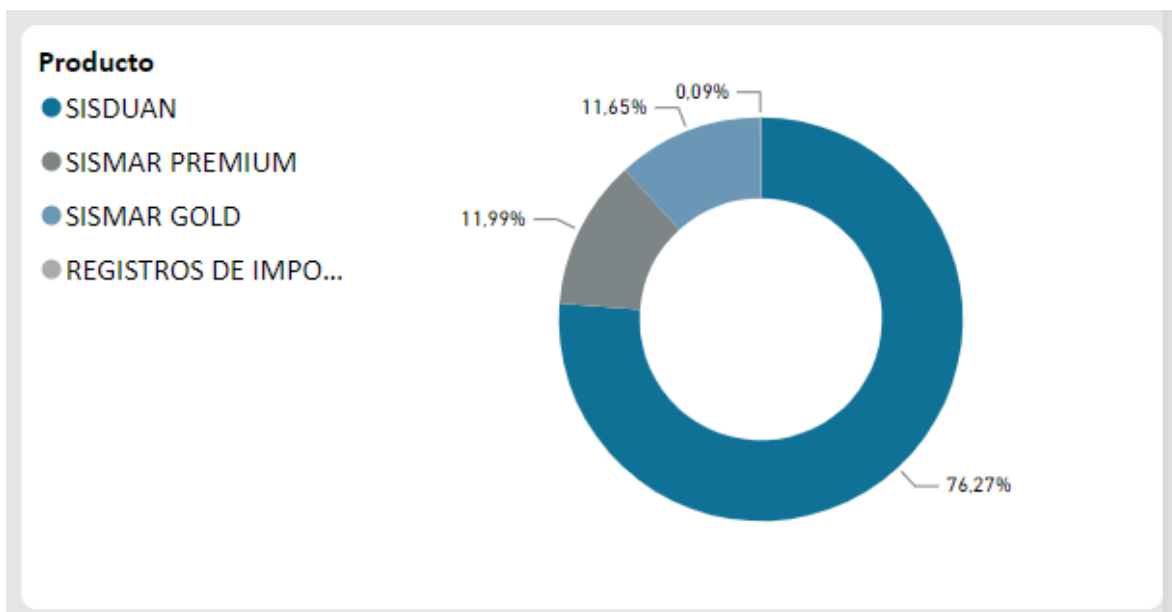
Participación de descarga de información por producto (SISMAR, SISDUAN, ETC)
(Ver Figura 29 y 30).

Figura 29 Power BI Participación Producto

producto	2023	Total
SISDUAN	105361	105361
SISMAR PREMIUM	16564	16564
SISMAR GOLD	16101	16101
REGISTROS DE IMPORTACION	124	124
Total	138150	138150

Fuente: Elaboración Propia

Figura 30 Power BI Participación Producto

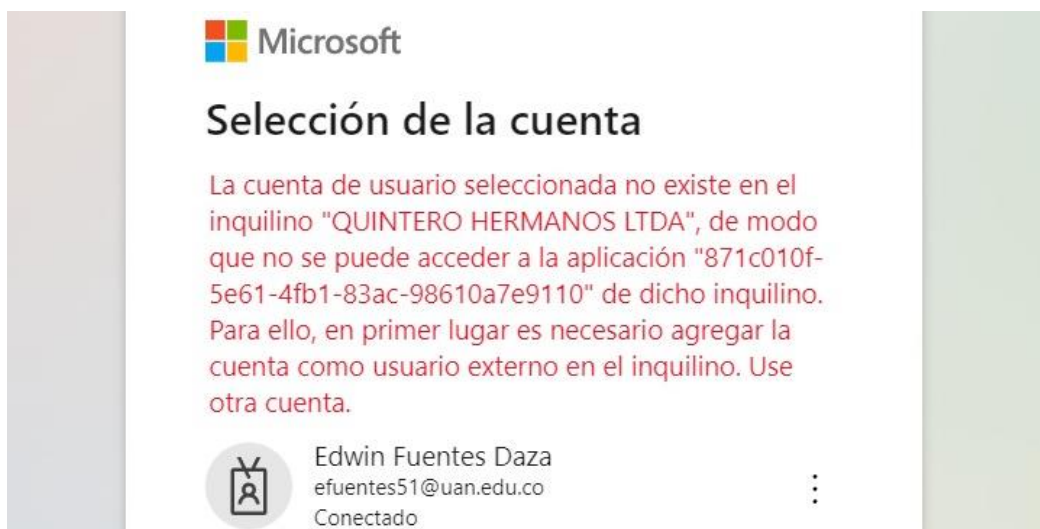


Fuente: Elaboración Propia

5.6.4 Pruebas al Dashboard

A continuación, se realizan diferentes pruebas, iniciando por el acceso al dashboard, donde efectivamente se valida que ningún otro usuario a excepción del coordinador comercial, puede tener acceso a dicha visualización (Ver Figura 31).

Figura 31 Revisión Acceso a Dashboard



Fuente: Elaboración Propia

Se valida que efectivamente se encuentre actualizando datos del repositorio sin ningún problema (Ver Figura 32).

Figura 32 Actualización Conjunto de Datos



Fuente: Elaboración Propia

Se publica el dashboard dentro del área de trabajo de reportes (Ver Figura 33).

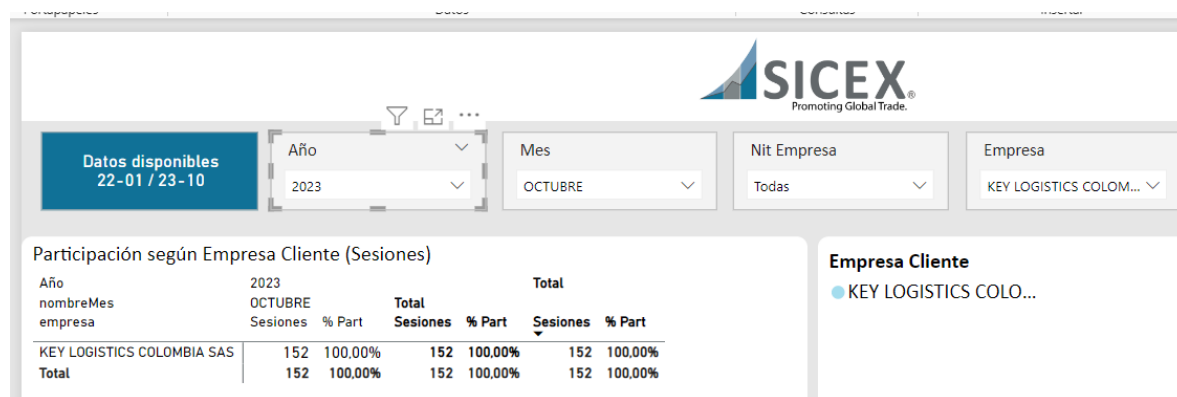
Figura 33 Publicación Exitosa Reporte



Fuente: Elaboración Propia

Se realiza validación de funcionamiento de filtros, con el fin de identificar que se estén aplicando correctamente y los datos filtrados correspondan a la realidad del repositorio de datos (Ver Figura 31).

Figura 34 Pruebas filtros Dashboard



Fuente: Elaboración Propia

6. Conclusiones

Se ha creado un Dashboard interactivo que permite a los miembros del equipo comercial explorar y analizar los datos sobre el uso del aplicativo web SICEX. Esto brinda a los usuarios una herramienta efectiva para comprender el comportamiento de los clientes y optimizar sus estrategias comerciales.

Se llevó a cabo una revisión de los registros almacenados en la base de datos semiestructurada, lo que proporcionó información valiosa sobre la forma en que se registran las actividades de los usuarios.

Se implementó un proceso ETL que transforma y carga los datos de manera efectiva en una bodega de datos estructurada, lo que facilita su acceso y análisis.

Los datos no estructurados del log de uso se transformaron en información estructurada y almacenada de manera organizada, lo que facilita su utilización por parte del equipo comercial.

El desarrollo del Dashboard, el análisis de la estructura de registros, el proceso ETL eficiente y la creación de la bodega de datos siguiendo la metodología de Kimball permiten proporcionar información detallada al equipo comercial.

El dashboard permitió al área comercial conocer el comportamiento de los usuarios en el aplicativo web SICEX, la cual se identificaron situaciones como:

Usuarios que no están utilizando dicho aplicativo: Al tener un conocimiento claro de los usuarios que no utilizan el aplicativo, el área comercial decide crear una estrategia para fidelizar a estos usuarios a través de capacitaciones, entrega de videos tutoriales y apoyo de parte de servicio al cliente para levantar información del por qué los usuarios no utilizan el aplicativo web.

Países que no se están consultando: Se identificó que existen países que tienen un bajo comportamiento de consulta por parte de los usuarios, el área comercial definirá y analizará con el área financiera y administrativa si es viable continuar o no, con estos países, ya que para obtener la información de cada país y publicarla en el aplicativo web generan altos costos.

Usuarios que tienen una mayor interacción con el aplicativo: Se identificaron usuarios que tienen mayor interacción con el aplicativo, permitiéndole al área comercial tener en cuenta a estos usuarios para levantar información sobre experiencias de usuario con cada uno de ellos y tenerlas en cuenta para la nueva versión del aplicativo web que se desarrollará el próximo año.

Mercancías más consultadas: Se identificó que existen mercancías importadas que son consultadas por usuarios que no están relacionados al core de su negocio, lo que da oportunidad al área comercial de crear una estrategia para ofrecer estudios especializados a estos usuarios que probablemente se encuentran investigando otros mercados

7. Recomendaciones

Algunas oportunidades para tener en cuenta a futuro son las siguientes:

Establecer un programa regular de mantenimiento proactivo para revisar y optimizar el código ETL. Esto incluye la identificación y corrección de problemas de rendimiento, así como la actualización de las transformaciones y cargas de datos para reflejar nuevos requisitos comerciales.

Establecer un proceso formal de gestión de cambios para evaluar y controlar cualquier modificación en las fuentes de datos, esquemas de bases de datos o requisitos comerciales. Esto garantiza que los cambios se gestionen de manera controlada y se documenten adecuadamente.

Mantener la seguridad de los datos en todo momento, lo cual implica cifrar la información sensible, gestionar los accesos de manera adecuada y protegerla contra posibles vulnerabilidades.

Implementar registros de auditoría para realizar un seguimiento de todas las operaciones ETL, lo que facilita la identificación de problemas y la trazabilidad de los datos.

Utilizar un sistema de control de versiones para el código ETL y la documentación. Esto ayuda a rastrear cambios a lo largo del tiempo y permite retroceder a versiones anteriores si es necesario.

8. Referencias

- Astera. (2023). Obtenido de <https://www.astera.com/es/type/blog/data-warehouse-concepts/>
- Cáceres, D. F. (2020). *Universidad Autónoma de Bucaramanga*. Obtenido de https://repository.unab.edu.co/bitstream/handle/20.500.12749/12049/20_20_Tesis_Diego_Fernando_Rangel.pdf?sequence=1&isAllowed=y
- CVN. (2023). Obtenido de <https://cvn.com.co/>
- DAMA. (2023). Obtenido de <https://www.dama.org/cpages/body-of-knowledge>
- DANE. (2023). Obtenido de <https://www.dane.gov.co/index.php/servicios-al-ciudadano/tramites/transparencia-y-acceso-a-la-informacion-publica/informacion-de-interes/preguntas-y-respuestas-frecuentes#:~:text=El%20Departamento%20Administrativo%20Nacional%20de,de%20la%20realidad%20soci>
- DANE. (2023). Obtenido de <https://www.dane.gov.co/index.php/servicios-al-ciudadano/tramites/transparencia-y-acceso-a-la-informacion-publica/informacion-de-interes/preguntas-y-respuestas-frecuentes#:~:text=El%20Departamento%20Administrativo%20Nacional%20de,de%20la%20realidad%20soci>
- DIAN. (2023). Obtenido de <https://www.dian.gov.co/dian/entidad/Paginas/Presentacion.aspx#:~:text=La%20Direcci%C3%B3n%20de%20Impuestos%20y,con%20la%20Direcci%C3%B3n%20de%20Aduanas>
- DIAN. (2023). Obtenido de <https://www.dian.gov.co/aduanas/Paginas/Declaracion-Especial-de-Importacion.aspx>
- Gonzalez, L. (06 de 07 de 2021). Obtenido de <https://explodat.cl/Analytics/business-intelligence/la-metodologia-kimball-para-data-warehouses-y-bi-exitosos/>
- Hernandez, J. P. (15 de 1 de 2023). *Universidad Oberta de Catalunya*. Obtenido de <https://openaccess.uoc.edu/bitstream/10609/147353/3/jguerreroheTFG0123memoria.pdf>
- International, D. (2017). *DAMA - DMBOK*. NEW JERSEY: Technics Publications.
- Johnna, C. (2023). *¿Qué es Power BI?* Obtenido de <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-power-bi.html>
- Legiscomex. (2023). Obtenido de <https://legiscomex.com/>
- Microsoft. (2023). Obtenido de <https://support.microsoft.com/es-es/topic/conceptos-b%C3%A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204>
- MySQL. (2023). Obtenido de <https://dev.mysql.com/doc/>

- Oracle. (2023). Obtenido de <https://www.oracle.com/co/database/what-is-a-data-warehouse/>
- Pandey Brij, S. E. (Septiembre de 2023). *Bulding ETL Pipelines With Python*. Obtenido de <https://www.segurosbolivar.com/blog/mundo-empresarial/4-retos-de-las-empresas-durante-la-pandemia>
- Pérez, C. (2018). *INTELIGENCIA DE NEGOCIOS - UN ENFOQUE PARA LA TOMA DE DECISIONES ESTRATÉGICAS EN LAS ORGANIZACIONES*. Obtenido de <https://www.itson.mx/oferta/mtin/SiteAssets/Paginas/productividad/Libro%20BI.pdf>
- Powerdata. (2023). Obtenido de <https://www.powerdata.es/integracion-de-datos>
- SAS. (2023). Obtenido de https://www.sas.com/es_co/insights/data-management/what-is-etl.html
- SICEX. (2023). Obtenido de <https://sicex.com/>

Anexos

Anexo 1 Diccionario de datos

DICCIONARIO DE DATOS		
TABLA	dimCalendario	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idCalendario	INT	Identificador único de registros en la tabla
fecha	DATE	Fecha de referencia
anio	SMALLINT	Año correspondiente a la fecha
trimestre	TINYINT	Trimestre al que pertenece la fecha
mes	TINYINT	Mes del año
nombreMes	VARCHAR(10)	Nombre del mes
diaMes	TINYINT	Día del mes
diaAnio	SMALLINT	Día del año
diaSemana	VARCHAR(10)	Día de la semana

TABLA	dimAccion	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idAccion	SMALLINT	Identificador único de registros de la tabla
accion	VARCHAR(50)	Descripción de una acción realizada por el usuario

TABLA	dimProducto	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idProducto	SMALLINT	Identificador único de registros en la tabla
producto	VARCHAR(50)	Descripción de un producto o ítem

TABLA	dimIntercambio	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idIntercambio	TINYINT	Identificador único de registros en la tabla
intercambio	VARCHAR(20)	Descripción del tipo de intercambio comercial

TABLA	dimPais	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idPais	SMALLINT	Identificador único de registros en la tabla
pais	VARCHAR (60)	Nombre del país
codigoISO	CHAR(3)	Código ISO del país

TABLA	dimEmpresa	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idEmpresa	INT	Identificador único de registros en la tabla
nitEmpresa	VARCHAR (16)	Número de Identificación Tributario de la empresa
empresa	VARCHAR (60)	Nombre de la empresa

TABLA	dimUsuario	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idUsuario	INT	Identificador único de registros en la tabla
nombres	VARCHAR(45)	Nombres del usuario
apellidos	VARCHAR(45)	Apellidos del usuario
eMail	VARCHAR(100)	Dirección del correo electrónico del usuario
telefono	VARCHAR(20)	Número de teléfono del usuario
idEmpresa	INT	Identificador de la empresa a la que pertenece el usuario

TABLA	dimCapitulo	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idCapitulo	SMALLINT	Identificador único de registros en la tabla
codCapitulo	CHAR(2)	Código del capítulo
capitulo	VARCHAR(200)	Descripción del capítulo

TABLA	dimPartida	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idPartida	SMALLINT	Identificador único de registros en la tabla
codPartida	CHAR(4)	Código de la partida
partida	VARCHAR(200)	Descripción de la partida

TABLA	dimSubpartida	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idSubpartida	SMALLINT	Identificador único de registros en la tabla
codSubpartida	CHAR(6)	Código de la subpartida
subpartida	VARCHAR(500)	Descripción de la subpartida

TABLA	sesionAplicacion	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
idsesion	INT	Identificador único de registros en la tabla
idUsuario	INT	Identificador del usuario
idCalendario	INT	Identificador de registro de calendario
cantidadSesion	INT	Cantidad de sesiones realizadas por el usuario

TABLA	monitoreoAplicacion	
CAMPO	TIPO DE DATO / LONGITUD	DESCRIPCION
monitoreoid	INT	Identificador único de registros en la tabla
idCalendario	INT	Identificador de registro de calendario
idProducto	SMALLINT	Identificador del producto
idAccion	SMALLINT	Identificador de la acción realizada
idUsuario	INT	Identificador del usuario
idPais	SMALLINT	Identificador del país
idIntercambio	TINYINT	Identificador del tipo de Intercambio
idCapitulo	SMALLINT	Identificador del capítulo
idPartida	SMALLINT	Identificador de la partida
idSubpartida	SMALLINT	Identificador de la subpartida
posicionArancelaria	VARCHAR(10)	Código de la posición arancelaria
cantRegistrosDescargados	INT	Cantidad de registros descargados según acción

Anexo 2 Script creación bodega de datos

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
```

```
-----
-- Schema monitoreoAppSicex
-----
```

```
-----
-- Schema monitoreoAppSicex
-----
```

```
CREATE SCHEMA IF NOT EXISTS `monitoreoAppSicex` DEFAULT CHARACTER SET
utf8 ;
USE `monitoreoAppSicex` ;
```

```
-----
-- Table `monitoreoAppSicex`.`dimPais`
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimPais` (
  `idPais` SMALLINT UNSIGNED NOT NULL,
  `pais` VARCHAR(60) NOT NULL,
  `codigoIso` CHAR(3) NOT NULL,
  PRIMARY KEY (`idPais`))
ENGINE = INNODB;
```

```
-----
-- Table `monitoreoAppSicex`.`dimProducto`
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimProducto` (
  `idProducto` SMALLINT UNSIGNED NOT NULL,
  `producto` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`idProducto`))
ENGINE = INNODB;
```

```
-----  
-- Table `monitoreoAppSicex`.`dimAccion`  
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimAccion` (  
  `idAccion` SMALLINT UNSIGNED NOT NULL,  
  `accion` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`idAccion`))  
ENGINE = INNODB;
```

```
-----  
-- Table `monitoreoAppSicex`.`dimIntercambio`  
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimIntercambio` (  
  `idIntercambio` TINYINT UNSIGNED NOT NULL,  
  `intercambio` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`idIntercambio`))  
ENGINE = INNODB;
```

```
-----  
-- Table `monitoreoAppSicex`.`dimCapitulo`  
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimCapitulo` (  
  `idCapitulo` SMALLINT UNSIGNED NOT NULL,  
  `codCapitulo` CHAR(2) NOT NULL,  
  `capitulo` VARCHAR(200) NOT NULL,  
  PRIMARY KEY (`idCapitulo`),  
  UNIQUE INDEX `codCapitulo_UNIQUE` (`codCapitulo` ASC))  
ENGINE = INNODB;
```

```
-----  
-- Table `monitoreoAppSicex`.`dimPartida`  
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimPartida` (  
  `idPartida` SMALLINT NOT NULL,  
  `codPartida` CHAR(4) NOT NULL,  
  `partida` VARCHAR(200) NOT NULL,  
  PRIMARY KEY (`idPartida`),  
  UNIQUE INDEX `codPartida_UNIQUE` (`codPartida` ASC))  
ENGINE = INNODB;
```

-- Table `monitoreoAppSicex`.`dimEmpresa`

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimEmpresa` (  
  `idEmpresa` INT NOT NULL,  
  `nitEmpresa` VARCHAR(16) NULL,  
  `empresa` VARCHAR(60) NULL,  
  PRIMARY KEY (`idEmpresa`))  
ENGINE = INNODB;
```

-- Table `monitoreoAppSicex`.`dimUsuario`

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimUsuario` (  
  `idUsuario` INT UNSIGNED NOT NULL,  
  `nombres` VARCHAR(45) NOT NULL,  
  `apellidos` VARCHAR(45) NOT NULL,  
  `eMail` VARCHAR(100) NULL,  
  `telefono` VARCHAR(20) NULL,  
  `idEmpresa` INT NOT NULL,  
  PRIMARY KEY (`idUsuario`),  
  INDEX `fk_dimUsuario_dimEmpresa_idx` (`idEmpresa` ASC),  
  CONSTRAINT `fk_dimUsuario_dimEmpresa`  
    FOREIGN KEY (`idEmpresa`)  
    REFERENCES `monitoreoAppSicex`.`dimEmpresa` (`idEmpresa`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = INNODB;
```

-- Table `monitoreoAppSicex`.`dimCalendario`

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimCalendario` (  
  `idCalendario` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `fecha` DATE NOT NULL,  
  `anio` SMALLINT UNSIGNED NOT NULL,  
  `trimestre` TINYINT UNSIGNED NOT NULL,  
  `mes` TINYINT UNSIGNED NOT NULL,  
  `nombreMes` VARCHAR(10) NOT NULL,  
  `diaMes` TINYINT UNSIGNED NOT NULL,  
  `diaAnio` SMALLINT UNSIGNED NOT NULL,  
  `diaSemana` VARCHAR(10) NOT NULL,
```

```
PRIMARY KEY (`idCalendario`),  
UNIQUE INDEX `fecha_UNIQUE` (`fecha` ASC)  
ENGINE = INNODB;
```

```
-----  
-- Table `monitoreoAppSicex`.`sesionAplicacion`  
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`sesionAplicacion` (  
  `idsesion` INT NOT NULL,  
  `idUsuario` INT UNSIGNED NOT NULL,  
  `idCalendario` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`idsesion`),  
  INDEX `fk_sesionAplicacion_dimUsuario1_idx` (`idUsuario` ASC),  
  INDEX `fk_sesionAplicacion_dimCalendario1_idx` (`idCalendario` ASC),  
  CONSTRAINT `fk_sesionAplicacion_dimUsuario1`  
    FOREIGN KEY (`idUsuario`)  
    REFERENCES `monitoreoAppSicex`.`dimUsuario` (`idUsuario`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_sesionAplicacion_dimCalendario1`  
    FOREIGN KEY (`idCalendario`)  
    REFERENCES `monitoreoAppSicex`.`dimCalendario` (`idCalendario`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = INNODB;
```

```
-----  
-- Table `monitoreoAppSicex`.`dimSubpartida`  
-----
```

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`dimSubpartida` (  
  `idSubpartida` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `codSubpartida` CHAR(6) NOT NULL,  
  `subpartida` VARCHAR(500) NOT NULL,  
  PRIMARY KEY (`idSubpartida`),  
  UNIQUE INDEX `codSubpartida_UNIQUE` (`codSubpartida` ASC))  
ENGINE = INNODB;
```

-- Table `monitoreoAppSicex`.`monitoreoAplicacion`

```
CREATE TABLE IF NOT EXISTS `monitoreoAppSicex`.`monitoreoAplicacion` (  
  `monitoreold` INT UNSIGNED NOT NULL,  
  `idCalendario` INT UNSIGNED NOT NULL,  
  `idProducto` SMALLINT UNSIGNED NOT NULL,  
  `idAccion` SMALLINT UNSIGNED NOT NULL,  
  `idUsuario` INT UNSIGNED NOT NULL,  
  `idPais` SMALLINT UNSIGNED NOT NULL,  
  `idIntercambio` TINYINT UNSIGNED NOT NULL,  
  `idCapitulo` SMALLINT UNSIGNED NOT NULL,  
  `idPartida` SMALLINT NOT NULL,  
  `idSubpartida` SMALLINT UNSIGNED NOT NULL,  
  `posicionArancelaria` VARCHAR(10) NOT NULL,  
  INDEX `fk_monitoreoAplicacion_dimProducto1_idx` (`idProducto` ASC),  
  INDEX `fk_monitoreoAplicacion_dimAccion1_idx` (`idAccion` ASC),  
  INDEX `fk_monitoreoAplicacion_dimPais1_idx` (`idPais` ASC),  
  INDEX `fk_monitoreoAplicacion_dimIntercambio1_idx` (`idIntercambio` ASC),  
  INDEX `fk_monitoreoAplicacion_dimCapitulo1_idx` (`idCapitulo` ASC),  
  INDEX `fk_monitoreoAplicacion_dimPartida1_idx` (`idPartida` ASC),  
  INDEX `fk_monitoreoAplicacion_dimSubpartida1_idx` (`idSubpartida` ASC),  
  INDEX `fk_monitoreoAplicacion_dimUsuario1_idx` (`idUsuario` ASC),  
  INDEX `fk_monitoreoAplicacion_dimCalendario1_idx` (`idCalendario` ASC),  
  CONSTRAINT `fk_monitoreoAplicacion_dimProducto1`  
    FOREIGN KEY (`idProducto`)  
    REFERENCES `monitoreoAppSicex`.`dimProducto` (`idProducto`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_monitoreoAplicacion_dimAccion1`  
    FOREIGN KEY (`idAccion`)  
    REFERENCES `monitoreoAppSicex`.`dimAccion` (`idAccion`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_monitoreoAplicacion_dimPais1`  
    FOREIGN KEY (`idPais`)  
    REFERENCES `monitoreoAppSicex`.`dimPais` (`idPais`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_monitoreoAplicacion_dimIntercambio1`  
    FOREIGN KEY (`idIntercambio`)  
    REFERENCES `monitoreoAppSicex`.`dimIntercambio` (`idIntercambio`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,
```

```
CONSTRAINT `fk_monitoreoAplicacion_dimCapitulo1`  
  FOREIGN KEY (`idCapitulo`)  
  REFERENCES `monitoreoAppSicex`.`dimCapitulo` (`idCapitulo`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_monitoreoAplicacion_dimPartida1`  
  FOREIGN KEY (`idPartida`)  
  REFERENCES `monitoreoAppSicex`.`dimPartida` (`idPartida`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_monitoreoAplicacion_dimSubpartida1`  
  FOREIGN KEY (`idSubpartida`)  
  REFERENCES `monitoreoAppSicex`.`dimSubpartida` (`idSubpartida`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_monitoreoAplicacion_dimUsuario1`  
  FOREIGN KEY (`idUsuario`)  
  REFERENCES `monitoreoAppSicex`.`dimUsuario` (`idUsuario`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_monitoreoAplicacion_dimCalendario1`  
  FOREIGN KEY (`idCalendario`)  
  REFERENCES `monitoreoAppSicex`.`dimCalendario` (`idCalendario`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = INNODB;  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

Anexo 3 ConexionBD.py

```
import mysql.connector

def mysql_cloud_sicex_r():
    cloud_config = {
        'host': 'pruebassicex-mysql.mysql.database.azure.com',
        'user': 'admin',
        'password': '+*****',
        'database': 'sicex_r',
    }
    connection = mysql.connector.connect(**cloud_config)
    return connection

def mysql_dw_oper():
    local_config = {
        'host': '172.16.1.70',
        'port': 3306,
        'user': 'consultas_prd',
        'password': '+++++++',
        'database': 'arancel_aduana',
    }
    connection = mysql.connector.connect(**local_config)
    return connection

def mysql_reporting():
    local_config = {
        'host': '172.16.1.34',
        'port': 3373,
        'user': 'powerBi',
        'password': '*****',
        'database': 'monitoreoAppSicex',
    }
    connection = mysql.connector.connect(**local_config)
    return connection
```


Anexo 4 ConexionCorreo.py

```
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart

# Configuración para el envío de correo
email_config = {
    'smtp_server': 'smtp.gmail.com',
    'smtp_port': 587, # Puerto SMTP
    'sender_email': 'efuentes51@uan.edu.co', # Tu dirección de correo electrónico
    'sender_password': 'Fuentes31leo', # Tu contraseña de correo
    'recipient_email': 'leofuentes1991@gmail.com', # Correo del destinatario
}

def enviar_correo(email_config, mensaje):
    try:
        # Crear el mensaje
        msg = MIMEMultipart()
        msg['From'] = email_config['sender_email']
        msg['To'] = email_config['recipient_email']
        msg['Subject'] = "Informe de Proceso" # Puedes personalizar el asunto

        # Agregar el mensaje al cuerpo del correo
        msg.attach(MIMEText(mensaje, 'plain'))

        # Conectar al servidor SMTP y enviar el correo
        server = smtplib.SMTP(email_config['smtp_server'],
            email_config['smtp_port'])
        server.starttls()
        server.login(email_config['sender_email'], email_config['sender_password'])
        server.sendmail(email_config['sender_email'],
            email_config['recipient_email'], msg.as_string())
        server.quit()

        print("Correo enviado con éxito.")

    except Exception as e:
        print(f"Error al enviar el correo: {str(e)}")

# Uso de la función para enviar un correo de error
error_message = "Ocurrió un error en el proceso."
enviar_correo(email_config, error_message)
```

Anexo 5 ETL.py

```
import mysql.connector
from ConexionBD import mysql_cloud_sicex_r, mysql_reporting, mysql_dw_oper
from ConexionCorreo import enviar_correo, email_config
```

```
try:
```

```
    # Conectarse al servidor en la nube
    cloud_connection = mysql_cloud_sicex_r()
    # Crear un cursor para ejecutar consultas
    cloud_cursor = cloud_connection.cursor()
```

```
    # Conectarse al servidor on premise reporting
    reporting_connection = mysql_reporting()
    # Crear un cursor para ejecutar consultas
    reporting_cursor = reporting_connection.cursor()
```

```
    # Conectarse al servidor on premise dw oper
    dwOper_connection = mysql_dw_oper()
    # Crear un cursor para ejecutar consultas
    dwOper_cursor = dwOper_connection.cursor()
```

```
    # GENERAR TABLA TEMPORAL DE MONITOREO EN MYSQL-CLOUD
    tabla_temporal = """
        CREATE TABLE tmp_monitoreo AS
        SELECT
            monitoreo_id AS monitoreold,
            monitoreo_producto_id AS monitoreoProductold,
            monitoreo_session AS monitoreoPais,
            monitoreo_accion AS monitoreoAccion,
            monitoreo_request AS monitoreoRequest,
            monitoreo_uinsert AS monitoreoUsuario,
            monitoreo_finsert AS monitoreoFecha,
            monitoreo_cantidad AS monitoreoCantidad
        FROM monitoreo
        WHERE monitoreo_finsert BETWEEN (SELECT
            DATE_FORMAT(DATE_SUB(CURRENT_DATE(), INTERVAL 1 MONTH), '%Y-%m-01'))
            AND
            (CONCAT(LAST_DAY(DATE_SUB(CURRENT_DATE(), INTERVAL 1 MONTH)), '
            23:59:59'))
        """
    cloud_cursor.execute("DROP TABLE IF EXISTS tmp_monitoreo")
    cloud_cursor.execute(tabla_temporal)
```

ACTUALIZACIÓN DE DIMENSIONES

DIMENSION CALENDARIO

```
sql = ""SELECT
    DATE_FORMAT(monitoreoFecha, '%Y-%m-%d') AS fecha,
    YEAR(monitoreoFecha) AS anio,
    QUARTER(monitoreoFecha) AS trimestre,
    MONTH(monitoreoFecha) AS mes,
    UPPER(MONTHNAME(monitoreoFecha)) AS nombreMes,
    DAY(monitoreoFecha) AS diaMes,
    DAYOFYEAR(monitoreoFecha) AS diaAnio,
    UPPER(DAYNAME(monitoreoFecha)) AS diaSemana
FROM tmp_monitoreo
GROUP BY 1""
cloud_cursor.execute("SET lc_time_names = 'es_ES';")
cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()
for res in resultado:
    fecha, anio, trimestre, mes, nombreMes, diaMes, diaAnio, diaSemana = res
    sql_insert = f""
        INSERT INTO dimCalendario (fecha, anio, trimestre, mes, nombreMes,
diaMes, diaAnio, diaSemana)
        VALUES ('{fecha}', '{anio}', '{trimestre}', '{mes}', '{nombreMes}', '{diaMes}',
'{diaAnio}', '{diaSemana}')
        ON DUPLICATE KEY UPDATE fecha = '{fecha}'
    ""
    reporting_cursor.execute(sql_insert)
# Realizar COMMIT en la base de datos Mysql Reporting
reporting_connection.commit()
```

DIMENSION ACCION

```
sql = "SELECT a.monitoreoAccion, b.accion_nombre FROM tmp_monitoreo AS a
LEFT JOIN accion AS b ON a.monitoreoAccion = accion_id GROUP BY 1;"
cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()
for res in resultado:
    id_accion, nombre_accion = res
    sql_insert = f""
        INSERT INTO dimAccion (idAccion, accion)
        VALUES ({id_accion}, '{nombre_accion}')
        ON DUPLICATE KEY UPDATE accion = '{nombre_accion}'
    ""
    reporting_cursor.execute(sql_insert)
```

DIMENSION PRODUCTO

```
sql = "SELECT a.monitoreoProductold, b.producto_nombre FROM
tmp_monitoreo AS a LEFT JOIN producto AS b ON a.monitoreoProductold =
producto_id GROUP BY 1"
cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()
for res in resultado:
    id_producto, nombre_producto = res
    sql_insert = f""""
        INSERT INTO dimProducto (idProducto, producto)
        VALUES ({id_producto}, '{nombre_producto}')
        ON DUPLICATE KEY UPDATE producto = '{nombre_producto}'
    """"
    reporting_cursor.execute(sql_insert)
```

DIMENSION INTERCAMBIO

```
sql = "SELECT intercambio_id, intercambio_nombre FROM intercambio WHERE
intercambio_id IS NOT NULL"
cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()
for res in resultado:
    id_intercambio, nombre_intercambio = res
    sql_insert = f""""
        INSERT INTO dimIntercambio (idIntercambio, intercambio)
        VALUES ({id_intercambio}, '{nombre_intercambio}')
        ON DUPLICATE KEY UPDATE intercambio = '{nombre_intercambio}'
    """"
    reporting_cursor.execute(sql_insert)
```

DIMENSION PAIS

```
sql = "SELECT a.monitoreoPais, b.pais_nombre, b.pais_uupdate AS pais_iso
FROM tmp_monitoreo AS a LEFT JOIN pais AS b ON a.monitoreoPais = pais_id
GROUP BY 1"
cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()
for res in resultado:
    id_pais, pais, iso_pais = res
    sql_insert = f""""
        INSERT INTO dimPais (idPais, pais, codigolso)
        VALUES ({id_pais}, '{pais}', '{iso_pais}')
        ON DUPLICATE KEY UPDATE pais = '{pais}', codigolso = '{iso_pais}'
    """"
    reporting_cursor.execute(sql_insert)
```

DIMENSION EMPRESA CLIENTE

```
sql = """SELECT c.empresa_id AS idEmpresa, c.empresa_nit AS nitEmpresa,
c.empresa_razon AS empresa
    FROM accesoprod AS a
    LEFT JOIN usuario AS b ON a.accesoprod_usuario_id = usuario_id
    LEFT JOIN empresa AS c ON b.usuario_empresa_id = c.empresa_id
    WHERE DATE_FORMAT(a.accesoprod_fdesactivacion, '%Y%m') >=
(SELECT DATE_FORMAT(DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1
MONTH), '%Y%m'))
    GROUP BY 1
    UNION
SELECT c.empresa_id AS idEmpresa, c.empresa_nit AS nitEmpresa,
c.empresa_razon AS empresa
    FROM tmp_monitoreo AS a
    LEFT JOIN usuario AS b ON a.monitoreoUsuario = usuario_id
    LEFT JOIN empresa AS c ON b.usuario_empresa_id = c.empresa_id
    GROUP BY 1
    UNION
SELECT c.empresa_id AS idEmpresa, c.empresa_nit AS nitEmpresa,
c.empresa_razon AS empresa
    FROM session_tokens AS a
    LEFT JOIN usuario AS b ON a.user_id = usuario_id
    LEFT JOIN empresa AS c ON b.usuario_empresa_id = c.empresa_id
    WHERE DATE_FORMAT(a.session_tokens_finsert, '%Y%m') = (SELECT
DATE_FORMAT(DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1 MONTH),
'%Y%m'))
    GROUP BY 1"""
cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()
for res in resultado:
    id_empresa, nit, empresa = res
    sql_insert = f"""
        INSERT INTO dimEmpresa (idEmpresa, nitEmpresa, empresa)
        VALUES ({id_empresa}, {nit}, {empresa})
        ON DUPLICATE KEY UPDATE nitEmpresa = '{nit}', empresa = '{empresa}'
    """
    reporting_cursor.execute(sql_insert)
```

DIMENSION USUARIOS

```
sql = """SELECT b.usuario_id, b.usuario_nombres AS nombres,
b.usuario_apellidos AS apellidos
, COALESCE(b.usuario_email, "") AS correo,
COALESCE(b.usuario_telefono, "") AS telefono, b.usuario_empresa_id AS
idEmpresa
FROM accesoprod AS a
LEFT JOIN usuario AS b ON a.accesoprod_usuario_id = usuario_id
WHERE DATE_FORMAT(a.accesoprod_fdesactivacion, '%Y%m') >=
(SELECT DATE_FORMAT(DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1
MONTH), '%Y%m'))
GROUP BY 1
UNION
SELECT b.usuario_id, b.usuario_nombres AS nombres,
b.usuario_apellidos AS apellidos
, COALESCE(b.usuario_email, "") AS correo,
COALESCE(b.usuario_telefono, "") AS telefono, b.usuario_empresa_id AS
idEmpresa
FROM tmp_monitoreo AS a
LEFT JOIN usuario AS b ON a.monitoreoUsuario = usuario_id
GROUP BY 1
UNION
SELECT b.usuario_id, b.usuario_nombres AS nombres,
b.usuario_apellidos AS apellidos
, COALESCE(b.usuario_email, "") AS correo,
COALESCE(b.usuario_telefono, "") AS telefono, b.usuario_empresa_id AS
idEmpresa
FROM session_tokens AS a
LEFT JOIN usuario AS b ON a.user_id = usuario_id
WHERE DATE_FORMAT(a.session_tokens_finsert, '%Y%m') = (SELECT
DATE_FORMAT(DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1 MONTH),
'%Y%m'))
GROUP BY 1"""
cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()
for res in resultado:
    id_usuario, nombres, apellidos, correo, telefono, id_empresa = res
    sql_insert = f"""
INSERT INTO dimUsuario (idUsuario, nombres, apellidos, eMail, telefono,
idEmpresa)
VALUES ({id_usuario}, '{nombres}', '{apellidos}', '{correo}', '{telefono}',
{id_empresa})
ON DUPLICATE KEY UPDATE nombres = '{nombres}', apellidos =
'{apellidos}', telefono = '{telefono}', eMail = '{correo}', idEmpresa = '{id_empresa}'
```

```
.....
```

```
reporting_cursor.execute(sql_insert)
```

DIMENSION CAPITULO

```
sql = "SELECT cod_capitulo, descripcion FROM capitulo"
```

```
dwOper_cursor.execute(sql)
```

```
resultado = dwOper_cursor.fetchall()
```

```
for res in resultado:
```

```
    cod_capitulo, desc_capitulo = res
```

```
    sql_insert = f"....."
```

```
        INSERT INTO dimCapitulo (codCapitulo, capitulo)
```

```
        VALUES ('{cod_capitulo}', '{desc_capitulo}')
```

```
        ON DUPLICATE KEY UPDATE codCapitulo = '{cod_capitulo}', capitulo =  
'{desc_capitulo}'
```

```
.....
```

```
reporting_cursor.execute(sql_insert)
```

DIMENSION PARTIDA

```
sql = "SELECT partida, descripcion FROM partida"
```

```
dwOper_cursor.execute(sql)
```

```
resultado = dwOper_cursor.fetchall()
```

```
for res in resultado:
```

```
    cod_partida, desc_partida = res
```

```
    sql_insert = f"....."
```

```
        INSERT INTO dimPartida (codPartida, partida)
```

```
        VALUES ('{cod_partida}', '{desc_partida}')
```

```
        ON DUPLICATE KEY UPDATE codPartida = '{cod_partida}', partida =  
'{desc_partida}'
```

```
.....
```

```
reporting_cursor.execute(sql_insert)
```

DIMENSION SUBPARTIDA

```
sql = "SELECT subpartida, descripcion FROM subpartida"
```

```
dwOper_cursor.execute(sql)
```

```
resultado = dwOper_cursor.fetchall()
```

```
for res in resultado:
```

```
    cod_subpartida, desc_subpartida = res
```

```
    sql_insert = f"....."
```

```
        INSERT INTO dimSubpartida (codSubpartida, subpartida)
```

```
        VALUES ('{cod_subpartida}', '{desc_subpartida}')
```

```
ON DUPLICATE KEY UPDATE codSubpartida = '{cod_subpartida}',
subpartida = '{desc_subpartida}'
.....
```

```
reporting_cursor.execute(sql_insert)
```

HECHOS SESIONES

```
sql = """SELECT user_id, DATE_FORMAT(session_tokens_finsert, '%Y-%m-%d') AS
fecha, COUNT(1) AS conteoSesion
FROM session_tokens
WHERE DATE_FORMAT(session_tokens_finsert, '%Y%m') = (SELECT
DATE_FORMAT(DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1 MONTH),
'%Y%m'))
```

```
GROUP BY 1,2
```

```
UNION
```

```
SELECT a.accesoprod_usuario_id AS user_id,
DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1 MONTH) AS fecha, 0 AS
conteoSesion
```

```
FROM accesoprod AS a
```

```
LEFT JOIN (SELECT DISTINCT user_id FROM session_tokens
```

```
WHERE DATE_FORMAT(session_tokens_finsert, '%Y%m') = (SELECT
DATE_FORMAT(DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1 MONTH),
'%Y%m')) AS b
```

```
ON a.accesoprod_usuario_id = b.user_id
```

```
WHERE DATE_FORMAT(a.accesoprod_fdesactivacion, '%Y%m') >=
(SELECT DATE_FORMAT(DATE_SUB(LAST_DAY(CURRENT_DATE()), INTERVAL 1
MONTH), '%Y%m'))
```

```
AND b.user_id IS NULL
```

```
GROUP BY 1
```

```
.....
```

```
cloud_cursor.execute(sql)
```

```
resultado = cloud_cursor.fetchall()
```

```
for res in resultado:
```

```
usuario_id, fecha, conteo_sesion = res
```

```
sql_insert = f"""
```

```
INSERT INTO sesionAplicacion (idUserio, idCalendario, cantidadSesiones)
```

```
SELECT '{usuario_id}', b.idCalendario ,{conteo_sesion}
```

```
FROM (SELECT '{fecha}' AS fecha) AS a
```

```
LEFT JOIN dimCalendario AS b ON a.fecha = b.fecha
```

```
LEFT JOIN sesionAplicacion AS c ON {usuario_id} = c.idUsuario AND
```

```
b.idCalendario = c.idCalendario
```

```
WHERE c.idCalendario IS NULL
```

```
.....
```

```
reporting_cursor.execute(sql_insert)
```


HECHOS MONITOREO

```
tabla_temporal = """
    CREATE TABLE `tmpMonitoreo` (
        `monitoreold` INT UNSIGNED NOT NULL DEFAULT '0',
        `monitoreoProductold` INT UNSIGNED DEFAULT NULL,
        `monitoreoPais` VARCHAR(32) NOT NULL,
        `monitoreoAccion` INT DEFAULT NULL,
        `monitoreoRequest` TEXT NULL,
        `monitoreoUsuario` INT UNSIGNED NOT NULL,
        `monitoreoFecha` DATETIME NOT NULL,
        `monitoreoCantidad` INT NOT NULL,
        PRIMARY KEY (monitoreold)

    )
"""

reporting_cursor.execute("DROP TABLE IF EXISTS tmpMonitoreo")
reporting_cursor.execute(tabla_temporal)

sql = """SELECT monitoreold, monitoreoProductold, monitoreoPais,
monitoreoAccion,
        monitoreoRequest, monitoreoUsuario, monitoreoFecha,
monitoreoCantidad
        FROM tmp_monitoreo
        """

cloud_cursor.execute(sql)
resultado = cloud_cursor.fetchall()

for row in resultado:
    sql_insert = f'INSERT INTO tmpMonitoreo VALUES ({", ".join(["%s"] *
len(row))})'
    reporting_cursor.execute(sql_insert, row)

# Realizar COMMIT en la base de datos Mysql Reporting
reporting_connection.commit()

# Uso de la función para enviar un correo de error
mensaje_email = (f"El proceso ETL se ha ejecutado satisfactoriamente")
enviar_correo(email_config, mensaje_email)

except mysql.connector.Error as error:
    # Manejar errores de conexión o consulta
    print(f"SE ENCONTRARON ERRORES: {error}")
    # Enviar correo de error
```

```
#error_message = f'Ocurrió un error: {str(e)}'  
# Uso de la función para enviar un correo de error  
mensaje_email = (f'El proceso ETL no se ejecutó debido a los siguientes  
errores: {error}')  
enviar_correo(email_config, mensaje_email)
```

finally:

```
# Cerrar el cursor y la conexión  
cloud_cursor.close()  
cloud_connection.close()  
reporting_cursor.close()  
reporting_connection.close()  
print("FINALIZADO")
```