Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

TEMA:

GENERACIÓN DE DATA WAREHOUSE PARA IMPLEMENTACIÓN DE BUSINESS INTELLIGENCE QUE PERMITA VISUALIZAR EL COMPORTAMIENTO DE LOS CLIENTES Y TOMAR MEDIDAS DE ACCIÓN COMERCIAL

AUTOR:

FRANCISCO XAVIER REYES MENA

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE

INFORMACIÓN MENCIÓN DATA SCIENCE

TUTOR: MSC. EDUARDO MONTERO

Quito, Julio – 2023

DEDICATORIA

A mis padres, por ser el motor para seguir avanzando y especializándome en esta área que tanto me ha llegado a gustar, a mi hermano por ser un ejemplo para seguir avanzando y superarme, a mi novia por su apoyo y sus consejos en todo momento.

AGRADECIMIENTO

En primer lugar, a Dios por permitirme seguir avanzando y superarme en todos los ámbitos de la vida, a la empresa AIRE.EC por brindarme las facilidades para llevar a cabo este proyecto, a mis profesores por brindarme su ayuda y conocimientos a lo largo de toda la maestría, a mi tutor por guiarme e impulsarme para que el proyecto se efectúe exitosamente.

RESUMEN

El presente trabajo tiene como finalidad el desarrollo de un *data warehouse,* que permita ser de ayuda para el área comercial en una empresa PYMES dedicada a la distribución de equipos de telecomunicaciones. En la actualidad que una empresa cuente con un sistema de *data warehouse* donde pueda centralizar sus datos, permite generar respuestas proactivas y agilizar los procesos involucrados con el área de ventas para la toma de decisiones. Respuestas a las preguntas tales como: ¿qué ítem se vende con mayor frecuencia, ¿cuál es la región con mayor cantidad de clientes?, son de vital importancia para establecer nuevos planes comerciales.

Por medio de información de ventas almacenada en formato CSV, desde el año 2014 al 2022, se ha generado un proceso de limpieza de datos que establece las variables relevantes, así como el formato adecuado. Además, se ha incurrido en la depuración de la información, ya que la misma contaba con errores producidos por ingreso manual. El archivo ha generado una tabla de hechos, la cual se cargó a un motor de visualización para generar una aplicación que despliegue la información relevante considerando: las ventas, los clientes, medidas de control y análisis multivariado.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I

1. Descripción Del Problema

1.1. Resumen ejecutivo

La finalidad del proyecto es crear un *data warehouse* que pueda ayudar al área comercial de una empresa PYMES, que se dedica a la distribución de equipos de telecomunicaciones. Actualmente, el contar con un *data warehouse* es determinante en una empresa, para poder centralizar los datos, y de esta manera generar respuestas proactivas y agilizar los procesos relacionados con el dominio de ventas para la toma de decisiones, respondiendo a preguntas tales como: ¿Qué producto se vende con más frecuencia? ¿Cuál es la región con mayor cantidad de clientes?, como ejemplo, las regiones con mayor número de clientes son determinantes en el desarrollo de nuevos planes de negocio.

1.2. Justificación

Actualmente en las PYMES se maneja información particionada debido a diferentes motivos, ya sea de gestión de información o de actualización de sistemas que impiden que se incurra en una unificación de la *data*, a esto se suman también inconvenientes en infraestructura. Todo esto genera que realizar el análisis de información sea un proceso complicado y de mucho tiempo.

El dar una pronta respuesta en una empresa; especialmente en aquellas que se enfocan en *retail*, implica que la directiva de ventas pueda tomar decisiones en tareas importantes como la tarea de pedidos, la cual es de suma importancia para poder realizar solicitudes a los proveedores, generando de esta manera que se tenga un producto con un aceptable nivel rotativo en la empresa. Además, se podría involucrar a diferentes áreas, como por ejemplo *marketing*, la cual se enfoca en campañas dirigidas y estratégicas para poder atacar un mercado dinámico.

La importancia de un *data warehouse* radica en poder generar en una organización, comprensión y usabilidad de los datos para tomar decisiones estratégicas.

1.3. Planteamiento del problema

La empresa aire.ec, tiene implementado el sistema ERP (*Quickbooks Enterprise Edition*) el cual ha sido de mucha ayuda al ser parte fundamental del negocio en áreas tales como: ventas, contabilidad, sistemas y bodega. Analizar la información de una base de datos en concreto no representa mayor problema, pero si se desea realizar análisis que involucren otras bases de datos o datos que se encuentran en otros documentos, este análisis conlleva mucho tiempo, y por lo general no es algo que se pueda revisar paulatinamente.

La mayoría de casos se presentan en el área comercial, por eso la problemática se expresa como la lenta respuesta del área comercial para definir comportamiento de clientes, especialmente por información segregada en diferentes fuentes de datos (archivos excel y bases de datos del sistema *ERP* principal).

Problema

- Lenta respuesta del área comercial para definir comportamiento de clientes

Causa

- Información
    - Segregada
    - En diferentes formatos
    - Con inconsistencias
- Planificación
    - Tiempos de ejecución no reales
    - Comunicación entre áreas
- Consulta de datos
    - La seguridad con la que cuenta la empresa, hace que el proceso de obtención de información de una base antigua sea largo y burocrático

- Personal externo para el área de servidores donde se almacenan los datos y la seguridad computacional, lo que genera una dependencia del tiempo de respuesta en función de la disponibilidad del personal

1.4. Contextualización del tema u objeto

En el 2000 ya se mencionaba la fuerte importancia de un *data warehouse* para mostrar cómo el "manejo de un sistema de información puede ayudar a mejorar el desempeño de un departamento de una Universidad Privada" (Castillo Montaño, 2000).

Hay que tomar en cuenta que "Un *data warehouse* exitoso es aquel que se convierte en parte integral en la búsqueda de la información que requiere la empresa. Y de esta forma, ayuda a tomar las decisiones que permitan alcanzar las metas de la organización. Sin embargo, se debe tomar en cuenta que un *data warehouse* no es un producto de *software* o *hardware* específico, sino un conjunto de componentes y procesos alineados a las estrategias de la organización." (Castillo Montaño, 2000).

En el 2021 ya se ha abordado la importancia de un *data warehouse* para instituciones de estado, y se concluye que "la inteligencia de negocios aplicada sirvió para ayudar a la toma de decisiones gracias a la rápida obtención de resultados a través de los cuales pudieron identificarse patrones o tendencias de comportamiento en situaciones determinadas" (Moffa, 2015).

En el 2022 se presentó una tesis que se enfoca en el desarrollo de un *data warehouse* en un sistema *ERP* mediante la metodología Hefesto, en esta "se demostró que la implementación de un modelo dimensional en un gestor de base de datos mejoró el tiempo de acceso a los datos, este es debido a la aplicación de la metodología de Inteligencia de Negocio e indicadores de gestión" (Castro Jara, 2022).

1.5. Objetivos

1.5.1. Objetivo General

Generar *Data warehouse* para implementar *Business Intelligence* mediante la unificación de información segregada que permita visualizar el comportamiento de los clientes y tomar medidas de acción comercial en una PYME de telecomunicaciones en Quito-Ecuador.

1.5.2. Objetivos Específicos

- Extraer la información del sistema principal, para unificarlo y generar un archivo .csv de carga.
- Transformar la información para generar una *data* depurada.
- Aplicar técnicas de *visual analytics* en el análisis de los datos.
- Emplear la metodología de Ralph Kimball para el desarrollo de *DataWarehouse*.

CAPÍTULO II

2. Marco Teórico

2.1. Data warehouse

2.1.1. Definición de Data warehouse

Un *data warehouse* (almacén de datos), es un sistema que tiene como propósito habilitar y dar soporte a las tareas de inteligencia empresarial (*business intelligence*). La mayoría de *data warehouses* se han creado para tareas de consulta y análisis, debido a esto, almacenan gran cantidad de información la cual en su mayoría proviene de datos históricos de diferentes fuentes.

Gracias a sus capacidades analíticas, las organizaciones pueden obtener información empresarial valiosa a partir de los datos y mejorar las decisiones. Con el tiempo, se construye un registro histórico de gran valor para los expertos en datos y los analistas de negocio. Gracias a estas funciones, un almacén de datos puede considerarse la "fuente única de datos" de una organización (Oracle).

2.1.2. Tipos de Data warehouse

Para los *data warehouse* se definen 3 tipos:

Offline, que se genera por una actualización programada, la cual puede ser diaria, semanal o mensual, y donde los datos se almacenan en una estructura integrada, para que los demás puedan acceder a ella y llevar a cabo la presentación de informes (Tecnologías Información).

En tiempo real, el cual presenta una actualización constante a medida que se generan nuevos datos.

Integrado, son aquellos que pueden ser utilizados por otros sistemas, hay que mencionar que también se pueden integrar con otros *data warehouse* en busca de conectar información relevante.

2.1.3. Impacto e importancia del análisis de datos en empresas

En la investigación empírica doctoral de Gonzales López (Gonzales López, 2012) aplicada en Perú, se proyecta estimar el impacto que tienen la *Data Warehouse* (DW) y la Inteligencia de Negocios (BI) en el desempeño de las empresas en un país en vías de desarrollo. La investigación recoge el criterio de todos los *stakeholders* en *data warehouse* y *business intelligence* dentro de una empresa, ya que cuenta con los criterios que se dan, desde la gerencia que desarrolla este tipo de producto y comentarios de los usuarios directos. De esta investigación se destaca: la calidad de la información, el uso del sistema, calidad del sistema, calidad del servicio y satisfacción del usuario como los principales constructos que generan un impacto en el desempeño de *data warehouse* y *business intelligence*.

En el documento de licenciatura de García Cortez (García Cortez, 2022), se presenta el paradigma del *data warehouse* y su uso con el fin de mostrar crecimiento, evolución y optimización de los diferentes datos que conllevan a los sistemas complejos. La investigación presenta como reto la migración de datos a un *data warehouse*, tomando en cuenta la preservación del presupuesto en todo el flujo del proyecto. De esta manera, se presenta que la ventaja principal radica en las estructuras de almacenamiento de la información, ya que puede ser almacenada por modelos estrella, copos de nieve, cubos relacionales, entre otros.

En el documento Treviño Reyes, Rivera Rodríguez y Garza Alonso (Treviño Reyes, Rivera Rodríguez, & Garza Alonso, 2020) se genera énfasis en la importancia del análisis de datos, y cómo estos representan la llave

principal para la permanencia en el mercado empresarial actual. Con este estudio se busca dar a conocer las ventajas competitivas que el análisis de datos puede generar para las empresas. Como principales ventajas se menciona: la mejora de la gestión empresarial, agilidad en la toma de decisiones y la vinculación de clientes a través del *marketing*.

En la investigación de García Pérez (García Pérez, 2020), se hace referencia a la inteligencia de negocios como la habilidad de transformar los datos en información y conocimiento que permita una adecuada toma de decisiones. Además, se enfoca en el análisis de datos, dirigido a disponibilizar datos con calidad y oportunidad. Algo a destacar de esta investigación es la generación de desarrollos internos / locales, que busquen generar en las empresas sistemas y medidas que ayuden al tratamiento de la información, principalmente cuando un factor determinante sea el alto costo de herramientas externas.

2.1.4. Proyectos de éxito en Data warehouse con beneficio a empresas PYMES

En la investigación realizada por Espinoza y Sotelo (Sánchez Espinoza & Canelo Sotelo, 2019) en Perú, se establece la problemática orientada a consultas de índole empresarial en las PYMES, en ésta se define que llegan a ser poco eficientes, con lo cual, el proyecto busca mejorar los tiempos de respuesta con optimización de almacenamiento. Para las PYMES se establece que resultados en formato *excel* tienen mucha acogida por la facilidad y generalización de uso del sistema.

En la investigación de Merino y Merino (García Merino & García Merino, 2018) se estudia los modelos de *business intelligence*. Sin embargo, también se establece la importancia de los *data warehouse* como punto de partida para la integración de la información que luego será referente para el

análisis. En esta investigación se puede observar que es reducida la cantidad de PYMES que cuentan con especialistas de TICs, lo que conlleva a que, proyectos de *data warehouse* no tengan el interés apropiado y por lo tanto no se incurra en el desarrollo.

En la tesis elaborada por Alicia Amadora (Zas Vázquez, 2022) se genera un *data warehouse* para una empresa ficticia de telecomunicaciones. En el proyecto se ha especificado las bondades del tratamiento de información y sobre todo la importancia de un adecuado proceso *ETL* para obtención de datos de buena calidad para responder a preguntas analíticas del negocio. En el proyecto también se establece que para que el *data warehouse* se mantenga óptimo, es importante que se realice mantenimientos, los cuales, no solo verifican el funcionamiento del mismo, sino que también incurren en el ingreso de nueva *data* de calidad.

En la tesis de Castro Jara (Castro Jara, 2022) se expresa la importancia de los *data warehouse* por ser útil en el filtrado y procesamiento de los datos que generan estructuras, con las cuales se obtiene una ayuda en el procesamiento y la visualización para mejorar el proceso de toma de decisiones. La aplicación de metodología Hefesto permitió establecer objetivos claros en cada fase del proyecto. Una acotación importante es que al momento de llevar a cabo estos proyectos se debe tener en cuenta las versiones de los sistemas, buscando que las mismas sean compatibles y no generen conflicto.

2.2. Metodologías de construcción para Data warehouse

Existen varías metodologías, la cuales han abierto campo en la construcción de un *Data warehouse;* y que facilitan su despliegue y comprensión. Cabe mencionar que cada una de ellas presenta diferencias relevantes en su diseño, mantenimiento,

elementos, etc (Mendoza, 2022). Entre las más relevantes se cuenta con: Ralph Kimball, Bill Inmon y Dan Linsted.

2.2.1. Metodología Ralph Kimball

Para Kimball, un *Data warehouse* es la copia de los datos transaccionales que se encuentran específicamente estructurados para generar informes y consultas analíticas que sean de apoyo a la toma de decisiones (Kimball & Ross, 2013).

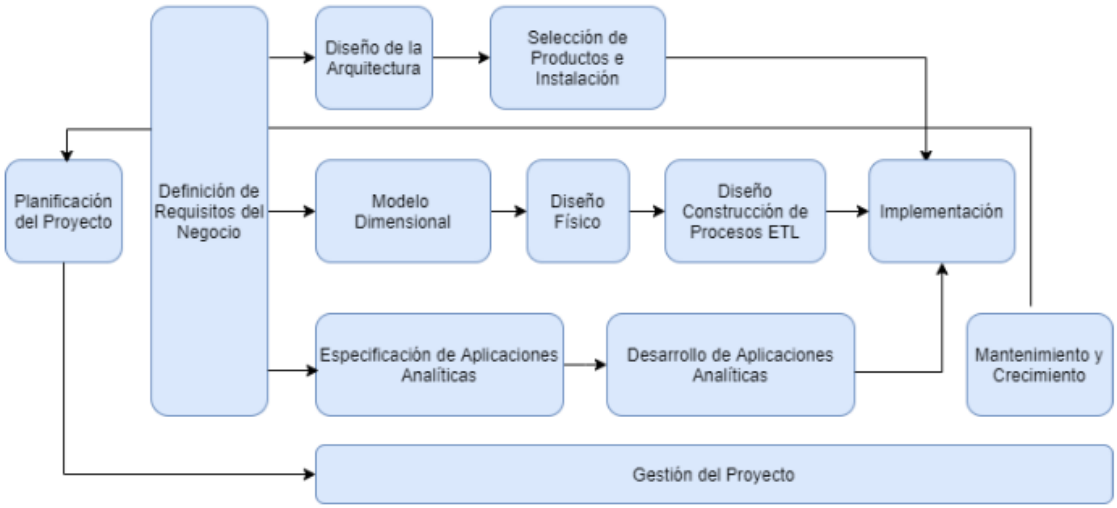


Figura 1 Metodología Kimball

Fuente: (Forero Castañeda & Sánchez García, 2022)

La metodología cuenta con las siguientes características:

- Se enfoca en el negocio, ya que se inicia con la definición de requisitos del negocio, con lo cual se enfoca en la creación de relaciones sólidas para generar valor.
- Su base de información es integrada, fácil de gestionar y genera un amplio rendimiento en el cual se puede apreciar rápidamente los requerimientos del negocio.
- Cada proceso consta de tareas bien definidas, lo que permite establecer incrementos significativos, que no sean extensos

ni cortos, de manera similar a una metodología ágil (Garcia & Rodas Silva, 2022).

- Cuenta con menor flexibilidad que la metodología *Bill Inmon* al momento de realizar una modificación.

2.2.2. Metodología Bill Inmon

Para asegurar un marco lógico de los datos, esta metodología coloca en el centro de la información corporativa al *Data warehouse* del cual se generarán a posteriori los diferentes *Data marts* específicos (Arias López, Molina García, & Sáenz Osorio, 2022).

Un *Data marts* se entiende como un sistema de almacenamiento de información, el cual contiene datos específicos de una unidad de negocio en una organización (Amazon).

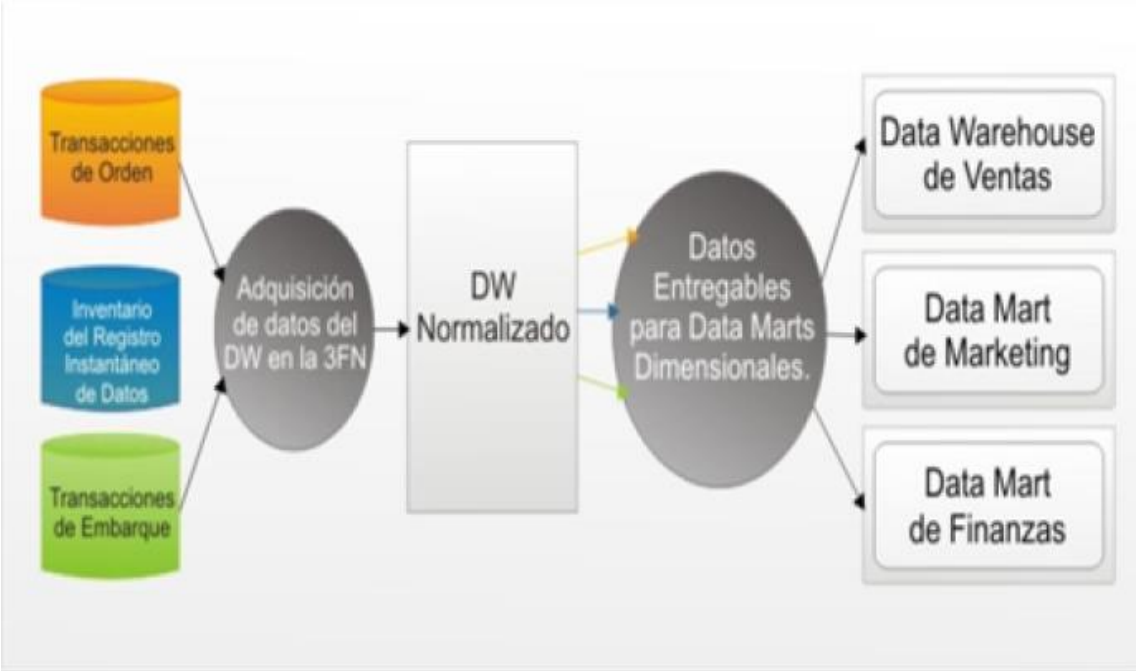


Figura 2 Metodología Bill Inmon

Fuente: (Garcia & Rodas Silva, 2022)

La metodología cuenta con las siguientes características:

- Todos los cambios en el tiempo, quedan registrados, de esta manera los informes y los análisis generados presentan versionamiento.
- La metodología establece no volatilidad, con lo cual la información no se modifica ni tampoco se elimina, una vez que la información es ingresada esta, solamente puede ser leída.
- Presenta una mayor flexibilidad a los cambios que se puedan presentar (ingreso de una nueva fuente de datos, nueva necesidad del negocio, nuevo proceso analítico).
- Por su complejidad, esta metodología requiere de una mayor capacidad para el modelado y el almacenamiento.

2.2.3. Metodología Dan Linsted

También llamada metodología *Data Vault*, está metodología es útil cuando las empresas presentan una cantidad exponencial y constante de datos (flexibilidad constante), lo que genera que se presenten problemas de rediseño y mantenimiento.

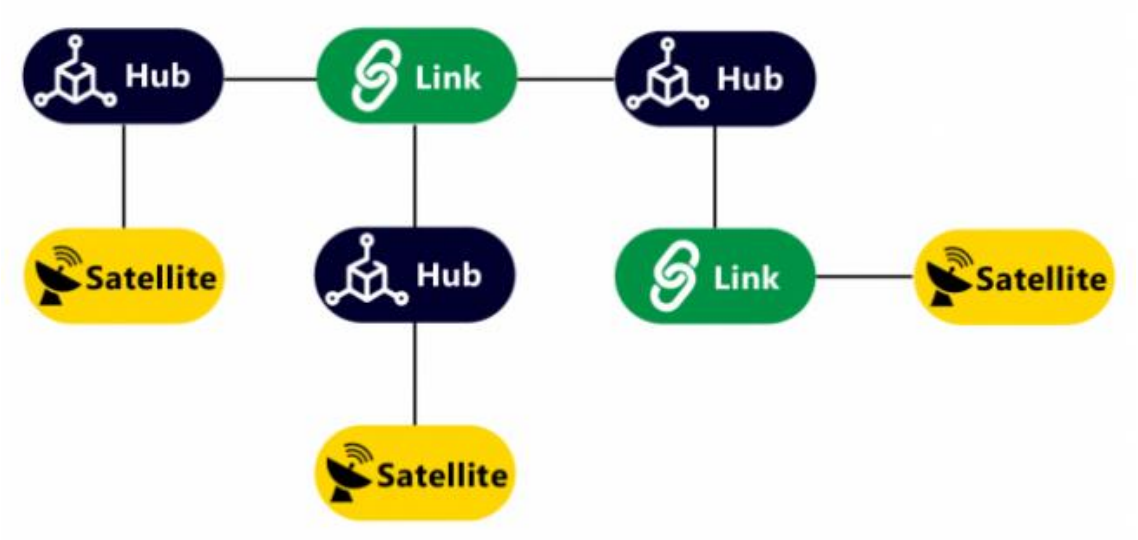


Figura 3 Metodología Dan Linsted

Fuente: (Mendoza, 2022)

Está metodología hace uso de 3 elementos principales:

- Hub: Contiene las claves únicas de un caso particular, como por ejemplo HUB_EMPLEADO.
- Link: Genera un rastreo de las relaciones que se dan en todos los *hubs*, como por ejemplo HUB_EMPLEADO_TIENDA
- Satélite: Presenta todos los atributos relacionados con *links* o *hubs* y se encarga de mantenerlo actualizado.

La metodología cuenta con las siguientes características:

- Permite que la automatización de procesos *ETL* sea más sencilla.
- Ideal cuando se presentan cambios frecuentes con las fuentes de datos, ya sea por datos nuevos o por la agregación de una nueva fuente de datos.

- Por su estructura, permite que se pueda generar un rastreo de la información, lo cual puede llegar a ser útil para temas de auditoría de datos.
- A diferencia de las otras dos metodologías, está es relativamente nueva, con lo cual no se cuenta con un sustento amplio de documentación que ayude en el despliegue de la misma, generando así un mayor esfuerzo para la exploración y adaptación.

2.3. Selección de Herramientas

2.3.1. Procesos ETL (Extraction Transform and Load)

Los procesos enfocados en la extracción, transformación y carga de la información en sistemas de destino, son relevantes, principalmente para generar un proceso de limpieza en los datos.

Los procesos *ETL* permiten bondades tales como: canalización flexible y adaptable de datos, reducción de errores por medio de automatización, entre otras (Naeem, 2020).

2.3.1.1. Lenguaje Python

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el *machine learning* (*ML*). Los desarrolladores utilizan *Python* porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software *Python* se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo (Amazon).

Python cuenta con librerías específicas para la ciencia de datos, entre estas tenemos: *pandas*, *numpy*, *scikit-learn*, entre otras. Entre la variedad de aplicaciones tenemos la creación y la gestión de estructura de datos, el despliegue

visual de la información y la generación de procesos de extracción, transformación y carga.

2.3.1.2. Lenguaje R

R es un entorno de software libre y lenguaje de programación interpretado, es decir, ejecuta las instrucciones directamente, sin una previa compilación del programa a instrucciones en lenguaje máquina. El término entorno, en R, se refiere a un sistema totalmente planificado y coherente. Este entorno es comúnmente utilizado para la computación estadística y gráfica, ya que dispone de una amplia variedad de técnicas estadísticas (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, agrupamiento, etc.) y gráficas. Funciona en plataformas UNIX y sistemas similares (incluidos FreeBSD y Linux), Windows y MacOS. Actualmente su desarrollo es responsabilidad del *R Development Core Team*. Forma parte de un proyecto colaborativo y abierto donde los usuarios pueden publicar paquetes que extienden su configuración básica (repositorio oficial de paquetes) (UNIR, 2019).

R cuenta con varios paquetes orientados a la ciencia de datos, entre ellos: paquetes R, códigos reproducibles de R, *Ggplot*, entre otros. R puede manejar de mejor manera proyectos de ciencia de datos orientados a temas estadísticos que *python*, ya que su finalidad se especifica en el análisis de datos. Sin embargo, no cuenta con una sintaxis tan amigable, con lo cual su curva de aprendizaje puede ser mayor que *python*.

2.3.1.3. Jupyter

El proyecto de código abierto *Jupyter* pretende generar una plataforma computacional que permita utilizar diferentes lenguajes. El pilar en el que se sustenta es la interfaz web *Jupyter Notebook*, que aglutina la ejecución de código, inclusión de texto junto a ecuaciones en *LaTeX* (vía MathJax), video, y todo lo que se pueda visualizar con un navegador. Su funcionalidad, por otro lado, puede

ser extendida gracias a herramientas como *nbconvert*, encargada de gestionar la conversión a formatos como *LaTeX*, *PDF* o presentaciones web basadas en *Reveal.js*, o *ipywidgets* que proporciona *widgets* interactivos dentro del documento. Su uso ha crecido en popularidad debido a su gran flexibilidad, fácil acceso a través del navegador y las posibilidades añadidas de separar el propio documento del núcleo de cálculo en el lenguaje escogido (Díaz García & Cabrera Granado, 2018).

2.3.2. Visualización de datos

En 1973, Francis Anscombe realizó una publicación relevante a la cual denominó "cuarteto de Anscombe", en esta manifestaba el por qué los datos siempre deben ser representados de manera gráfica antes de ser analizados. La investigación se llevó a cabo con cuatro conjuntos de datos, los cuales contenían estadísticos muy similares. Sin embargo, después de un análisis visual se podía observar que, a pesar de tener estadísticas iguales, la data se comportaba de una manera distinta.

La importancia de la visualización de datos es simple: ayuda a las personas a ver, interactuar y comprender mejor los datos. Ya sea simple o compleja, la visualización correcta puede atraer a todos a la misma página, independientemente de su nivel de experiencia (Tableau).

2.3.2.1. *Power BI by Microsoft*

Power BI es un servicio gratuito de análisis de negocio basado en la nube y visualización de datos de negocio. Esta herramienta está incluida en la suite de productividad *Microsoft Office 365* y permite controlar la salud de un negocio mediante un *dashboard* en vivo, crear informes interactivos y acceder a los datos en cualquier lugar con las aplicaciones nativas de móvil, es decir, aplicaciones que funcionen sobre sistema operativos *iOS* y *Android* (Arimetrics).

2.3.2.2. Tableau

Tableau permite hacer un análisis de los datos en un tiempo corto, mediante la generación de visualizaciones y demostraciones impactantes. Posee una configuración flexible, debido a que se puede desplegar bajo un servidor, de manera local o en la nube. Destaca por su facilidad para integrar datos de diferentes orígenes y su sencillez de uso, permitiendo realizar un análisis ágil y rápido en un entorno colaborativo (Neteris).

2.3.2.3. Qlik Sense

Qlik Sense es una aplicación avanzada de visualización de datos que mediante una interfaz interactiva permite que cualquier persona de un equipo de datos pueda generar con facilidad visualizaciones flexibles, interactivas, que impulsen la exploración y el descubrimiento. Se puede generar cuadros de mando a medida, mediante la conexión de múltiples fuentes de información, teniendo la capacidad de personalizarlos sin límites para que reflejen todo lo relevante para una empresa (Bitec).

CAPÍTULO III

3. Metodología

3.1. Administración y planeación del proyecto

3.1.1. Definición del proyecto

Con la importancia que han tomado en la actualidad los entornos como *data warehouse*, e identificando las bondades de las empresas que lo usan con respecto a la competencia, se ha generado la necesidad de reunir en un único entorno, toda la data necesaria para consultas; esperando de esta manera, generar una aceleración en la toma de decisiones, y con la factibilidad de que las mismas estén fundamentadas en los históricos de la empresa.

3.1.2. Alcance del proyecto

El proyecto se presenta como un desarrollo piloto, enfocado en primera instancia en todo el ámbito de ventas, incurriendo en productos, cantidades, valores, provincias y clientes. El proyecto se presentará en una herramienta de despliegue visual, con lo cual se busca que sea intuitivo y de fácil entendimiento para la gerencia.

3.1.3. Requerimientos del negocio

En una empresa PYMES orientada a la venta de equipos de telecomunicaciones, es de importancia contar con un proyecto de esta índole, que permita:

- Entender la información histórica almacenada en diferentes archivos del *ERP* central (*Quickbooks Enterprise*).

- Acelerar los tiempos de respuesta a peticiones asociadas con el área de ventas.
- Generar visualizaciones intuitivas para que la gerencia pueda comprender de manera rápida los movimientos del área de ventas.
- Encontrar *insigtht* relevantes para el área de ventas

3.2. Fases del proyecto

En esta sección se detallan los procesos de extracción, transformación y carga (*ETL*), y la recolección de los datos. Se revisan todos los procesos desde la recolección de los datos hasta la generación del *data warehouse*.

Figura 4 Preparación y carga de Datos

Fuente: Elaboración propia

3.2.1. Recolección de datos

Los datos se han obtenido de una empresa PYMES orientada a la venta de equipos de telecomunicaciones, la cual es propietaria de la

información y ha brindado la misma para la elaboración de este proyecto. Los datos que se examinarán en el proyecto corresponden al periodo comprendido entre los años 2014 y 2022.

El formato en el cual se ha realizado la recolección de datos es "Comma Separated Values" (CSV) y la exportación final de la limpieza se realiza en el mismo formato.

3.2.2. Preparación de los datos

Para la preparación de los datos se ha incurrido en la unificación de varios archivos CSV, los cuales se han obtenido de las ventas por año a detalle del sistema principal *Quickbooks*. Una verificación rápida de los archivos se ha generado con el sistema *Office Excel,* de esta manera se ha podido constatar una totalidad de 108 variables iniciales y una cantidad de 87423 instancias.

3.2.3. Limpieza de datos

Para la limpieza y depuración de los datos se ha utilizado *Google Colab* con lenguaje de programación *python.* Se ha unificado los archivos y se ha desarrollado transformaciones que generen datos con una estructura común, cabe mencionar que tanto para los *outliers* como para las inconsistencias que se encontraron se aplicó reglas específicas brindadas por el contador de la empresa, el cual conoce a profundidad esta información.

3.2.4. Diseño del data warehouse

El *data warehouse* contará con una **tabla de hechos** especificado en las siguientes variables: 5 continuas, 7 discretas, 25 nominales, 5 ordinales, la cual se detalla a continuación:

Tabla 1 Tabla de Hechos

Nombre	Tipo	Descripción
Trans #	Discreta	Maneja un número único por archivo generado de Quickbooks
Type	Nominal	Tipo de registro, el cual se divide en facturas y notas de crédito
Entered/Last Modified	Ordinal	Fecha de ingreso o modificación del registro
Last modified by	Nominal	Usuario de Quickbooks que ha realizado la última modificación
Date	Ordinal	Fecha de creación del registro
Num	Discreta	Número de documento del registro
Source Name	Nominal	Nombre del cliente
Name Address	Nominal	Dirección del cliente. Sin embargo, será adaptada internamente, ya que la misma no ha estado en uso
Name Street1	Nominal	Calle 1 de la dirección del cliente. Sin embargo, será adaptada internamente, ya que la misma no ha estado en uso
Name State	Nominal	Ubicación más exacta de la dirección. Sin embargo, será adaptada internamente, ya que la misma no ha estado en uso
Name Contact	Nominal	Nombre de contacto para despacho
Name Phone #	Discreta	Número de teléfono del contacto principal

Name E-Mail	Nominal	Email del cliente
Name	Nominal	Nombre del cliente, principalmente usado en caso que el cliente sea una empresa
Ship Date	Ordinal	Fecha de envío de la mercadería
Terms	Ordinal	Términos de pago
Due Date	Ordinal	Fecha de vencimiento del registro
Ruc:	Discreta	Número de identificación de los clientes
Email:	Nominal	Email del cliente, usado específicamente para facturación
Vendedor:	Nominal	Agente comercial encargado del cliente
Ciudad:	Nominal	Ciudad principal del cliente
Provincia:	Nominal	Provincia principal del cliente
Limite Credito	Discreta	Valor del límite de crédito para el registro
Documento Garantia	Nominal	Documento de garantía con el que cuenta el cliente, empleado principalmente para el tema de cobranza
Tipo ID	Nominal	Identificar de identificación, R usado para RUC y C para cédula
Contabilidad	Nominal	Especifica si el cliente debe llevar contabilidad, N usado para NO y S usado para SI
Item	Nominal	Item al que hace referencia el registro

Item Description	Nominal	Descripción del item al que hace referencia el registro
Account	Nominal	Cuenta interna de la empresa, usado para manejo del área contable
Sales Tax Code	Nominal	Identifica los registros que se han manejado con IVA o sin IVA
Qty	Discreta	Cantidad del item al que se hace referencia el registro
Sales Price	Continua	Precio de venta al que se hace referencia el registro
Credit	Continua	Valor enfocado en notas de crédito
Amount	Continua	Valor total del registro
Account Type	Nominal	Columna definida para la parte contable (venta de bienes y venta de mercadería)
Action	Nominal	Usado para tema de auditoría, identifica si el registro fue cambiado o no presentó cambio
Backordered	Discreta	Enfocado como valor negativo para las facturas anuladas
Ship To Address 1	Nominal	Dirección de envío
ITEM_FAMILIA	Nominal	Campo creado para obtener la familia del item
ITEM_SKU	Nominal	Campo creado para obtener el código único del item
LONGITUD	Continua	Campo creado con la longitud, hace referencia a la provincia
LATITUD	Continua	Campo creado con la latitud, hace referencia a la provincia

Fuente: Elaboración propia

3.3. Visualización

En esta sección se revisa todo lo relacionado con la presentación, específicamente la herramienta de visualización y los gráficos desarrollados enfocados en aspectos claves para el área de ventas.

Para la visualización, se trabajará con el sistema *PowerBI* debido a que presenta ventajas en la interacción de su interfaz, además que brinda planes accesibles para empresas PYMES. A continuación, se presenta como relevante en el cuadrante mágico de *Gartner ("Analytics and Business Intelligence Platforms").*

Figura 5 Cuadrante mágico de Gartner - Analytics and Business Intelligence Platforms

Fuente: (Gartner, 2023)

En la visualización de información, se ha incurrido en las siguientes gráficas:

3.3.1. Gráficos de barras

Este tipo de gráfico permite la comparación de valores numéricos, los cuales pueden ser números enteros o porcentajes. Puede mostrar comparación entre diferentes valores en subcategorías. Para casos agrupados, puede mostrar la eficacia de diferentes estrategias o métodos para lograr un objetivo (Tableau).

En la figura podemos observar que el ítem LOCOM5 representa la mitad de las ventas del producto LBE-M5-23, esto se debe a que este equipo fue muy demandado para comunicaciones *wireless.*

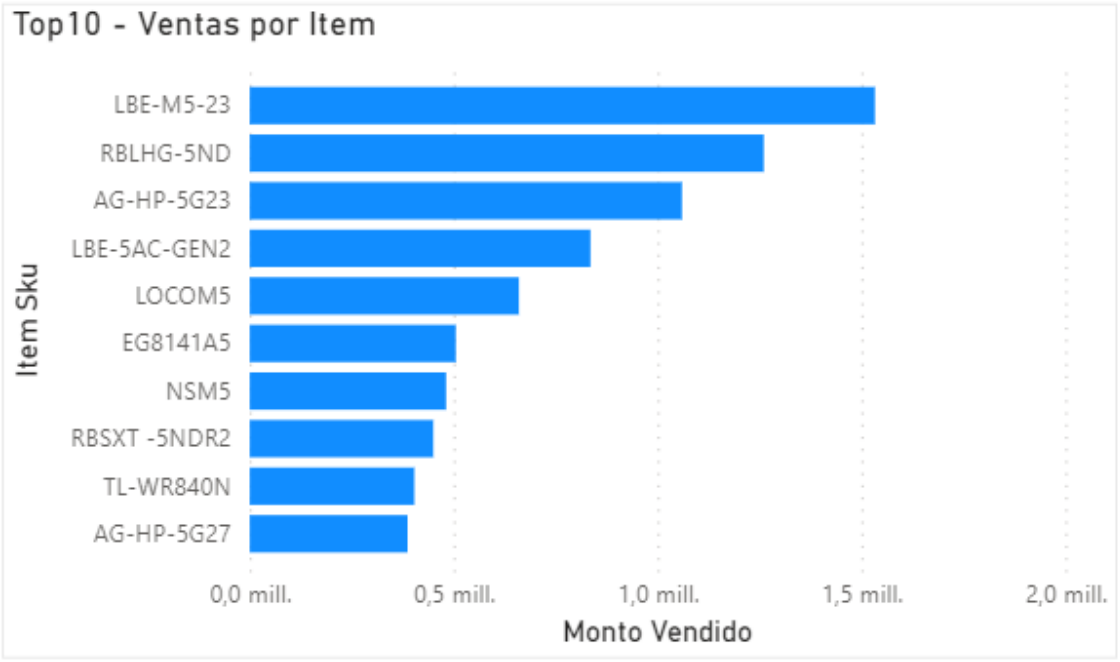


Figura 6 Gráfico de barras en *Power BI*

Fuente: Elaboración propia

3.3.2. Gráficos espaciales

Muestra la información usando su ubicación geográfica, lo que ayuda a establecer una ubicación que permite responder factores claves como: cantidad de ventas, controles logísticos en base a distancias, cantidad de clientes en una determinada región, entre otros (Tableau).

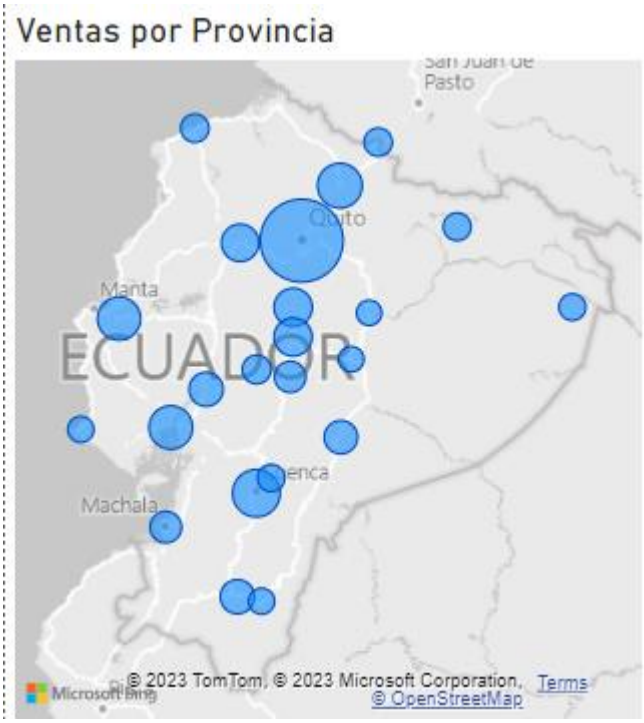


Figura 7 Gráfico espacial en *Power BI*

Fuente: Elaboración propia

3.3.3. Gráfico circular

Ayuda a mostrar la totalidad como un porcentaje, las partes del círculo brindan la representación de las categorías. Este gráfico es ideal cuando se poseen pocas categorías (**no más de 5**), ya que de ser así el gráfico puede llegar a generar una inadecuada interpretación. Además, se deben usar gráficos circulares para mostrar la relación de las diferentes partes con el

todo y funcionan mejor con dimensiones que tienen un número limitado de categorías. (Tableau).

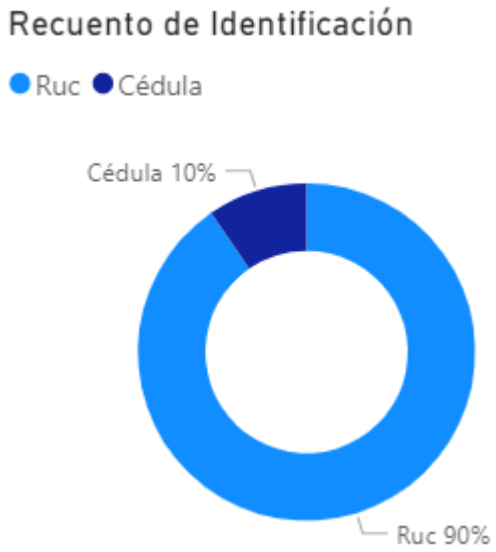


Figura 8 Gráfico circular en *Power BI*

Fuente: Elaboración propia

3.3.4. Gráfico de áreas

Los gráficos de área presentan la magnitud del cambio a lo largo del tiempo, con lo cual pueden ser usados para captar la atención en tendencias relevantes (Microsoft, 2022).

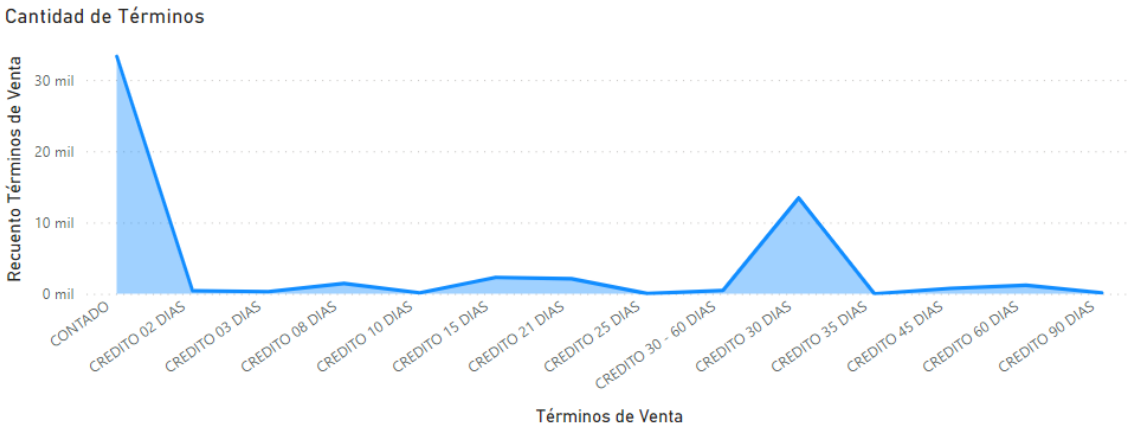


Figura 9 Gráfico circular en *Power BI*

Fuente: Elaboración propia

3.3.5. Gráfico de líneas

Este tipo de gráfico presenta valores secuenciales que ayudan a la identificación de tendencias. Además, es útil cuando se desea mostrar datos a lo largo del tiempo. Los gráficos de líneas también están enfocados en resaltar las diferencias y correlaciones dentro de los datos (Tableau).

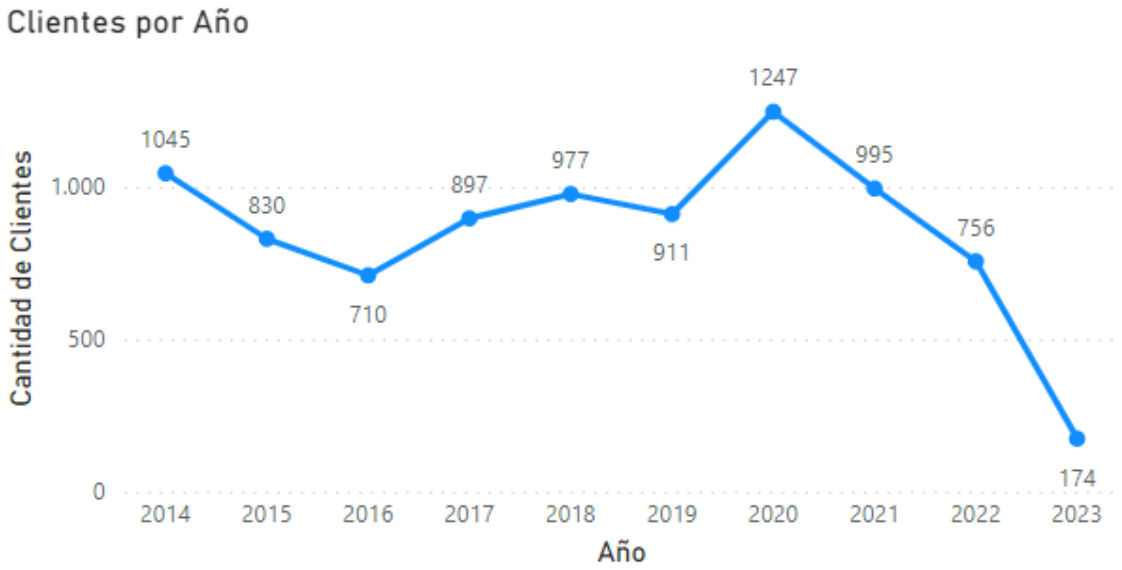


Figura 10 Gráfico de líneas en *Power BI*

Fuente: Elaboración propia

3.3.6. Gráfico de dispersión

Los gráficos de dispersión se usan para averiguar la intensidad de la relación entre dos variables numéricas. Se usan para responder a preguntas tales como: ¿cuál es la relación entre dos variables? ¿Cómo se distribuyen los datos? ¿Dónde están los valores atípicos? (ArcGIS Insights, 2023).

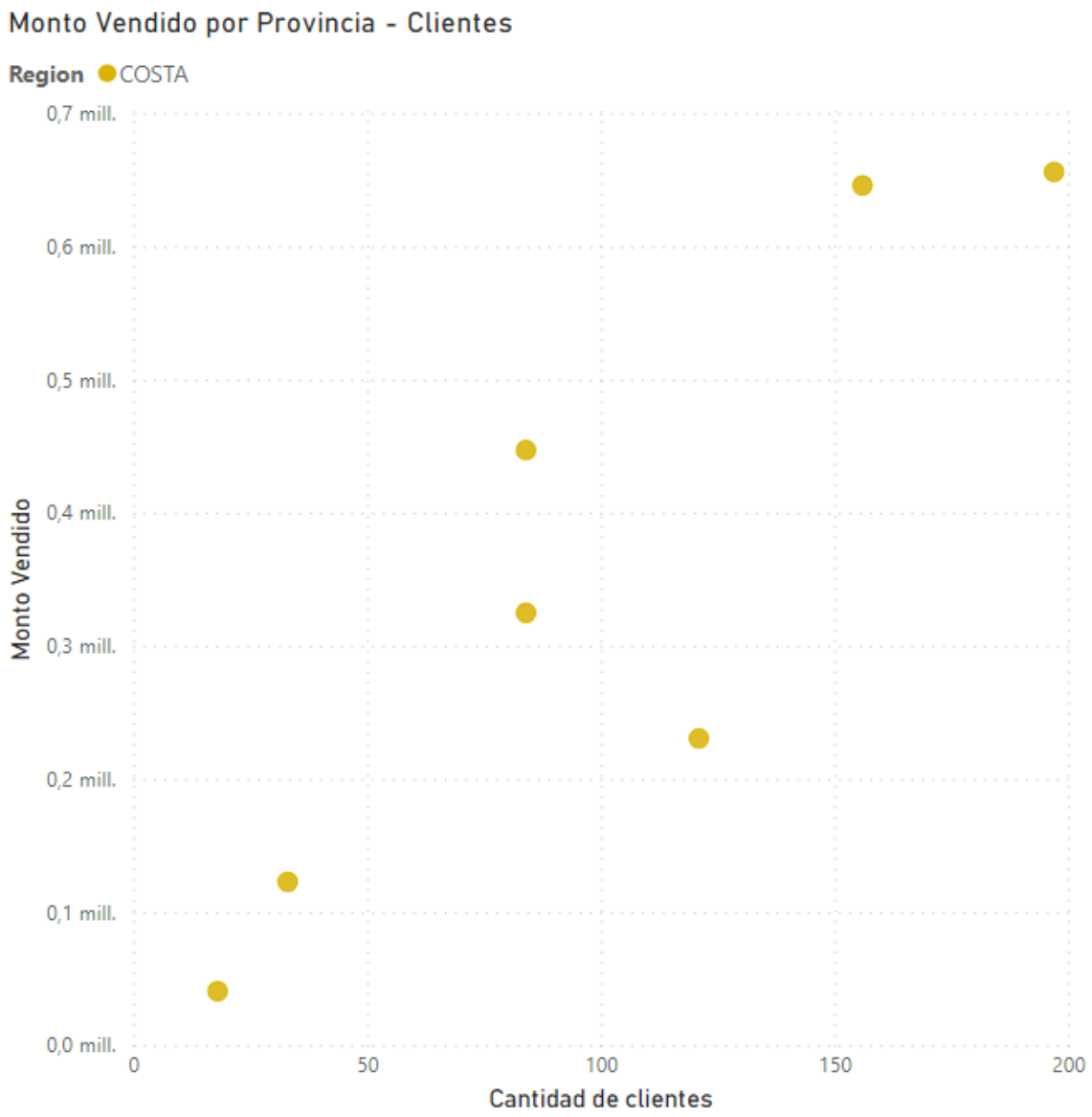


Figura 11 Gráfico de dispersión en *Power BI*

Fuente: Elaboración propia

3.4. Análisis de datos

El análisis de datos se ha dividido en cuatro secciones, y de las mismas se ha obtenido un análisis multivariado para determinar *Insights*.

Como resumen se presentan las siguientes tablas, de métricas y dimensiones.

Tabla 2 Tabla de Métricas

Métrica	Descripción	Pestaña de la app en *PowerBI*
Ventas	Se refiere al valor monetario o de cantidad que se ha generado	Ventas Clientes Análisis Multivariado
Clientes	Se refiere a la cantidad de clientes	Clientes Análisis Multivariado

Fuente: Elaboración propia

Tabla 3 Tabla de Dimensiones

Dimensión	Descripción	Pestaña de la app en *PowerBI*
Tiempo	Permite establecer día, mes y año de un determinado evento	Ventas Clientes Control Análisis Multivariado
Provincia	Establece el nombre del lugar para posicionar geográficamente	Ventas Clientes Análisis Multivariado
Familia	Establece una identificación interna del producto, en la mayoría de veces relacionado con la marca	Ventas
Producto	Hace referencia al ítem adquirido al proveedor	Ventas Análisis Multivariado
Vendedor	Maneja el código interno de los agentes de ventas	Ventas

Fuente: Elaboración propia

3.4.1. Overview

Presenta un resumen enfocado en las métricas de ventas más importantes, tales como:

- Monto vendido
- Total de facturación
- Total notas de crédito
- Cantidad vendida
- Familias de productos
- Cantidad de vendedores
- Cantidad de provincias alcanzadas

Este conjunto de métricas permite establecer un resumen clave para el área comercial y para la gerencia de la empresa.

Figura 12 Gráfico Overview en *Power BI*

Fuente: Elaboración propia

3.4.2. Ventas

Para la venta se maneja factores clave tales como:

- Monto vendido por mes y año

- Familia de producto
- Código de producto
- Monto por provincia
- Monto por vendedor

Encontrar y entender de manera rápida, ¿qué familia de productos se venden más?, ¿Cuál es el vendedor que está generando más ingresos?, ¿En cuál provincia debo potenciar las ventas?, entre otras. Establecen un amplio y claro panorama de nuevas acciones comerciales.

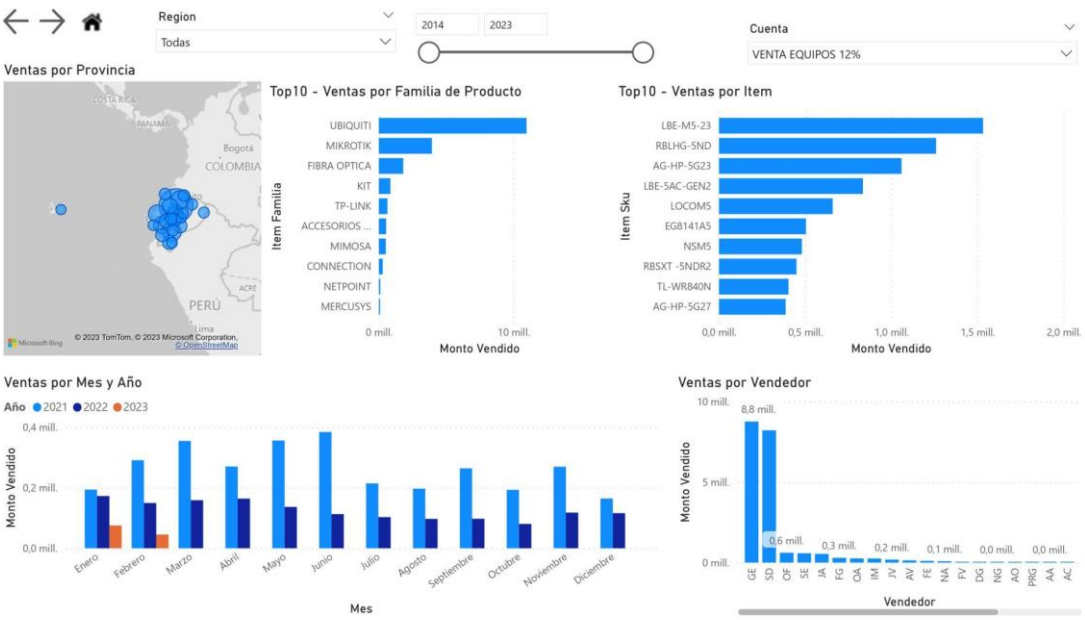


Figura 13 Gráfico Ventas en *Power BI*

Fuente: Elaboración propia

3.4.3. Clientes

Para la sección de clientes se ha determinado:

- Clientes con mayores ventas
- Recuentos de clientes por identificación (ruc o cédula)
- Cantidad de clientes por año

- Cantidad de términos de venta
- Cantidad de clientes por provincia

Establecer e identificar clientes con un alto monto de compras, permite determinar estructuras y reglas que contribuyan a mejorar y asegurar un mayor monto de venta, como, por ejemplo: mediante créditos que brinden beneficios a clientes mayoristas, así también, se podría desplegar promociones enfocadas en aumentar ventas y canalizar mayor fidelidad. Identificar rápidamente las provincias en donde se cuenta con más clientes, puede ayudar a generar planes de venta de forma regional, logrando una estabilidad ante determinadas temporadas en una región, como el caso de regreso a clases (aumento de cuota de internet) y la llegada del periodo invernal.

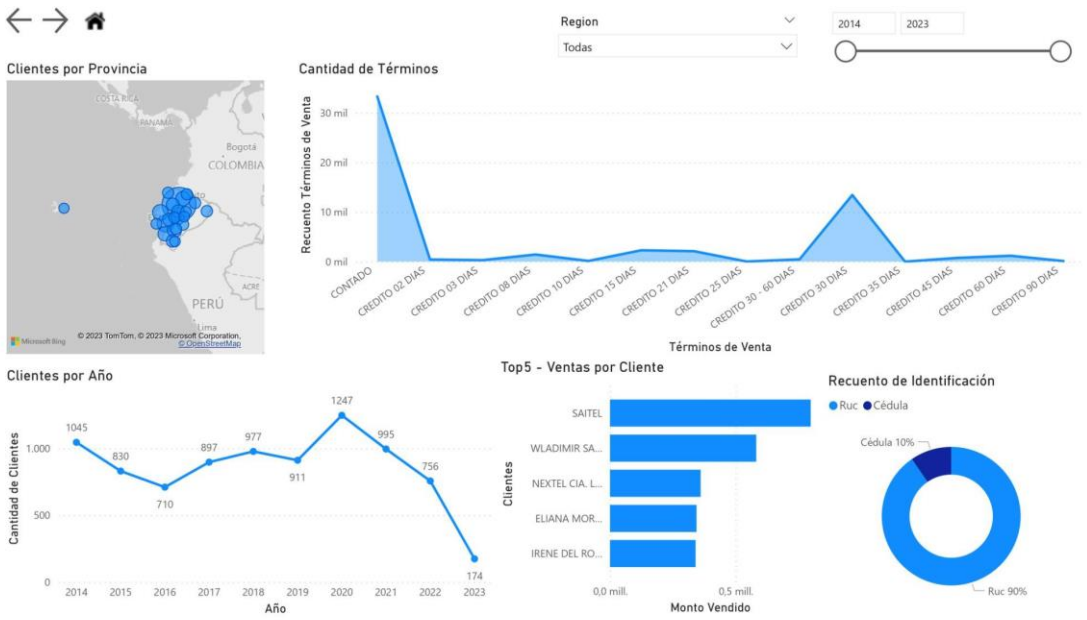


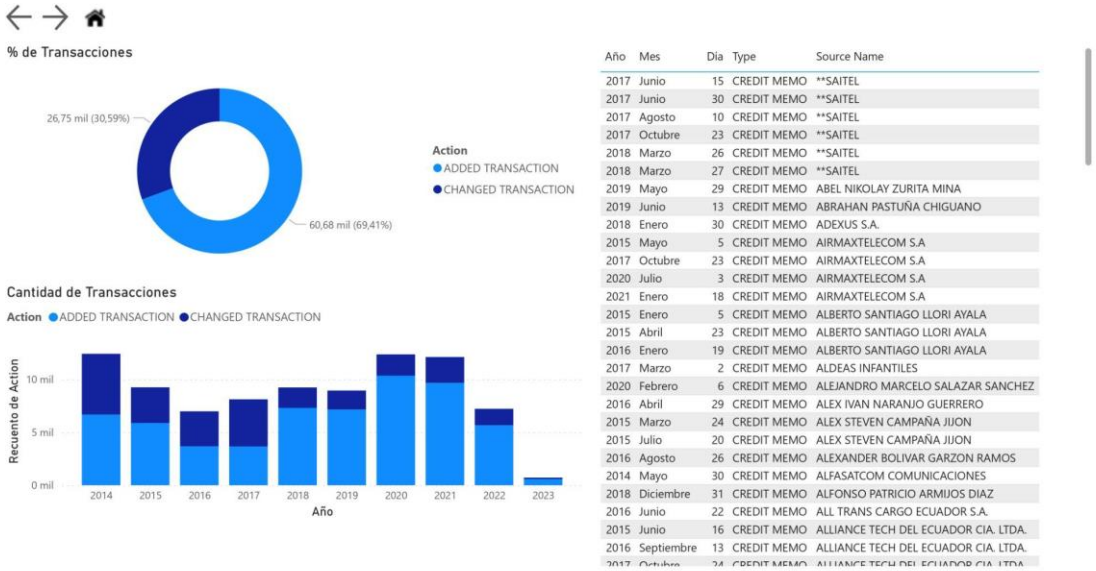
Figura 14 Gráfico Clientes en *Power BI*

Fuente: Elaboración propia

3.4.4. Control

Esta sección está enfocada en servir de ayuda para los procesos de auditoría, en donde de manera rápida se pueda evidenciar la cantidad de registros que han presentado cambios y aquellos que se han mantenido sin intervenciones.

Cabe mencionar que, si bien es una ayuda, el desarrollo de esta sección se la podría plantear como un proyecto a futuro para incrementar el *data warehouse* y generar un módulo específico para este fin.



Año	Mes	Dia	Type	Source Name
2017	Junio	15	CREDIT MEMO	**SAITEL
2017	Junio	30	CREDIT MEMO	**SAITEL
2017	Agosto	10	CREDIT MEMO	**SAITEL
2017	Octubre	23	CREDIT MEMO	**SAITEL
2018	Marzo	26	CREDIT MEMO	**SAITEL
2018	Marzo	27	CREDIT MEMO	**SAITEL
2019	Mayo	29	CREDIT MEMO	ABEL NIKOLAY ZURITA MINA
2019	Junio	13	CREDIT MEMO	ABRAHAN PASTUÑA CHIGUANO
2018	Enero	30	CREDIT MEMO	ADEXUS S.A.
2015	Mayo	5	CREDIT MEMO	AIRMAXTELECOM S.A
2017	Octubre	23	CREDIT MEMO	AIRMAXTELECOM S.A
2020	Julio	3	CREDIT MEMO	AIRMAXTELECOM S.A
2021	Enero	18	CREDIT MEMO	AIRMAXTELECOM S.A
2015	Enero	5	CREDIT MEMO	ALBERTO SANTIAGO LLORI AYALA
2015	Abril	23	CREDIT MEMO	ALBERTO SANTIAGO LLORI AYALA
2016	Enero	19	CREDIT MEMO	ALBERTO SANTIAGO LLORI AYALA
2017	Marzo	2	CREDIT MEMO	ALDEAS INFANTILES
2020	Febrero	6	CREDIT MEMO	ALEJANDRO MARCELO SALAZAR SANCHEZ
2016	Abril	29	CREDIT MEMO	ALEX IVAN NARANJO GUERRERO
2015	Marzo	24	CREDIT MEMO	ALEX STEVEN CAMPAÑA JIJON
2015	Julio	20	CREDIT MEMO	ALEX STEVEN CAMPAÑA JIJON
2016	Agosto	26	CREDIT MEMO	ALEXANDER BOLIVAR GARZON RAMOS
2014	Mayo	30	CREDIT MEMO	ALFASATCOM COMUNICACIONES
2018	Diciembre	31	CREDIT MEMO	ALFONSO PATRICIO ARMIJOS DIAZ
2016	Junio	22	CREDIT MEMO	ALL TRANS CARGO ECUADOR S.A.
2015	Junio	16	CREDIT MEMO	ALLIANCE TECH DEL ECUADOR CIA. LTDA.
2016	Septiembre	13	CREDIT MEMO	ALLIANCE TECH DEL ECUADOR CIA. LTDA.

Figura 15 Gráfico Control en *Power BI*

Fuente: Elaboración propia

3.4.5. Análisis multivariado

En esta sección se ha generado un análisis de las siguientes dimensiones y métricas:

- Monto de venta en base a la región y cantidad de clientes
- Monto de ventas en relación a la cantidad de clientes por año

Estos análisis presentan factores claves para la toma de decisiones, enfocado en establecer nuevas estrategias comerciales que puedan llegar a incrementar las ventas en una determinada región, y catalogar provincias en las cuales se presenten clientes que generen compras representativas.

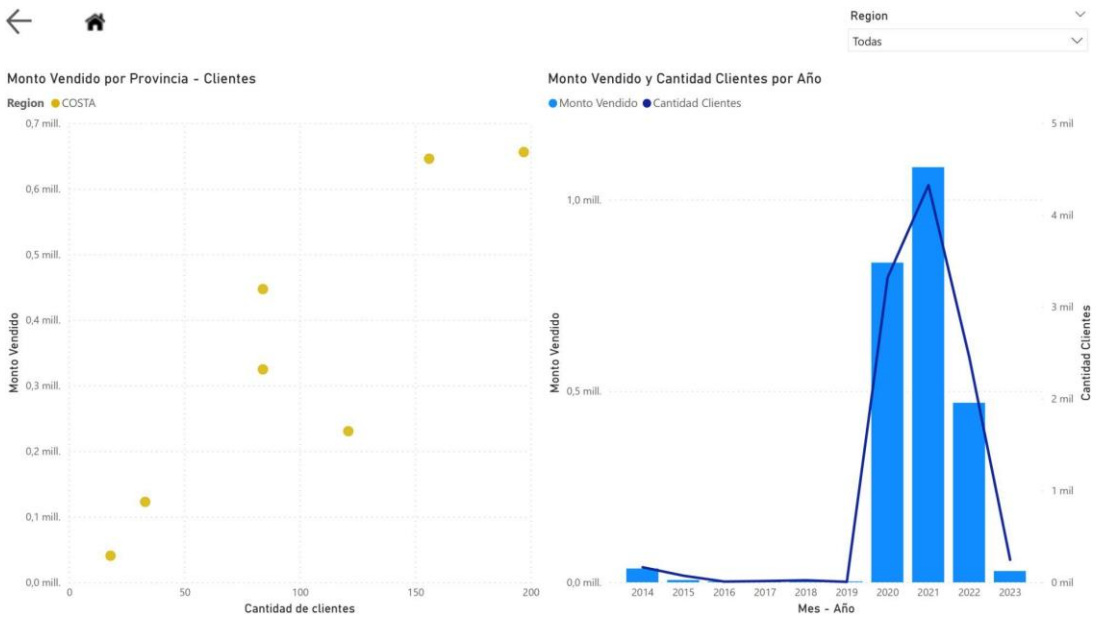


Figura 16 Gráfico Análisis multivariado en *Power BI*

Fuente: Elaboración propia

3.5. Validación y resultados

El modelo de datos resultante para el análisis fue el siguiente

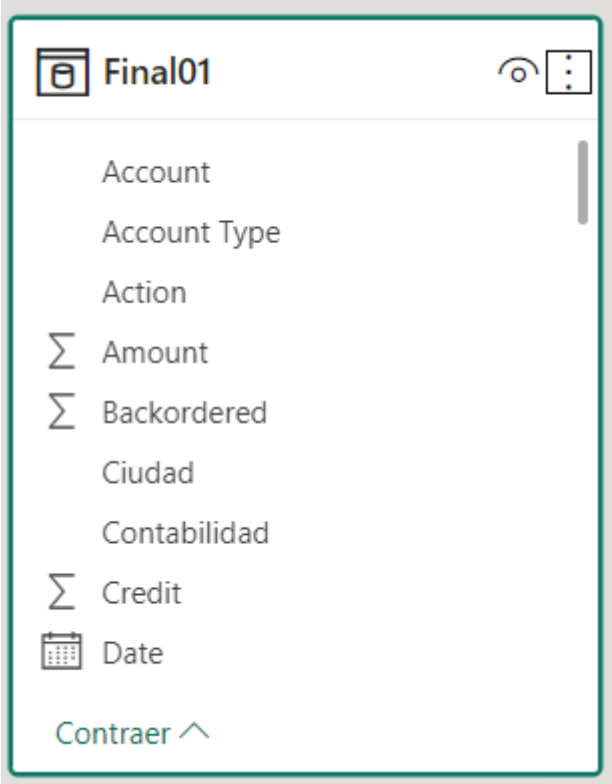


Figura 17 Tabla de Hechos resultante en *Power BI*

Fuente: Elaboración propia

Mediante los gráficos espaciales, se ha logrado determinar una correcta ubicación de los puntos de longitud y latitud, en vista que esta variable fue creada en el proceso de limpieza de datos, usando de referencia la provincia.

En el despliegue se determina una correcta división entre los ítems y la familia de los ítems, los cuales en el proceso de limpieza han sido creados del campo Item.

Se ha corroborado que cada variable se encuentre en el tipo de dato adecuado al momento de generar la exportación al documento CSV, de esta manera al ser importado en *PowerBI* no se ha incurrido en cambios drásticos de variables.

Mediante la generación de la tabla de hechos, y por medio de los filtros generados en las gráficas de *PowerBI* se destaca la rapidez y la fluidez del despliegue de las visualizaciones.

CAPÍTULO IV

4. Conclusiones y Recomendaciones

4.1. Conclusiones

El presente proyecto se desarrolló para brindar una solución a la toma de decisiones en el ámbito comercial de una empresa PYMES orientada a la venta de equipos de telecomunicaciones. Mediante el mismo, se ha logrado generar, de una manera más rápida criterios relevantes al momento de tomar decisiones comerciales, como, por ejemplo: ¿cuál fue el producto más vendido en un determinado periodo de tiempo?, ¿En qué provincia se ha tenido menos inserción?, ¿Cuáles son las provincias con mayor número de clientes?

Con la introducción del proyecto se ha visualizado la importancia y el valor de los datos para una empresa PYMES orientada a la venta de equipos de telecomunicaciones, generando de esta manera cabida a proyectos de depuración, limpieza y tratamiento de información. También se ha visto la necesidad de impulsar una normativa de datos que permita tener un control más exhaustivo.

A pesar que el proyecto está desarrollado para trabajar con la plataforma *Power BI,* esta no es una limitante para que el procesamiento generado pueda ser llevado a una aplicación más robusta si las necesidades y el presupuesto así lo disponen.

4.2. Recomendaciones

Como trabajo futuro se plantea el uso de herramientas propietarias que permitan generar una descarga de información de toda la base Quickbooks de una manera más automatizada. La misma se podría generar para vincular más tablas de hechos y dimensiones al proyecto.

Junto con el análisis de información, y una depuración de otras áreas de la empresa, como, por ejemplo: logística, auditoría, importaciones, entre otras, se podría generar proyectos enfocados a más ámbitos de datos, como la inteligencia artificial con aplicaciones enfocadas a los algoritmos de *machine learning*.

Para que el proyecto siga aportando valor a la empresa, es de importancia que la empresa PYMES orientada a la venta de equipos de telecomunicaciones destine recursos, tanto técnicos como de presupuesto para el mantenimiento y actualización de la información.

BIBLIOGRAFÍA

Amazon. (s.f.). *¿Qué es Python?* Obtenido de https://aws.amazon.com/es/what-is/python/

Amazon. (s.f.). *¿Qué es un data mart?* Obtenido de https://aws.amazon.com/es/what-is/data-mart/#:~:text=Un%20data%20mart%20es%20un,sistema%20de%20almacenamiento%20m%C3%A1s%20grande.

ArcGIS Insights. (2023). *Crear y utilizar un gráfico de dispersión.* Obtenido de https://doc.arcgis.com/es/insights/latest/create/scatter-plot.htm

Arias López, C. A., Molina García, N. U., & Sáenz Osorio, V. M. (Enero de 2022). *Diseño de un modelo dimensional para soportar el proceso de negocios de steam.* Obtenido de https://ri.ues.edu.sv/id/eprint/27474/1/Dise%C3%B1o%20de%20un%20modelo%20dimensional%20para%20soportar%20el%20proceso%20de%20negocios%20de%20Steam.pdf

Arimetrics. (s.f.). *Qué es Power BI.* Obtenido de https://www.arimetrics.com/glosario-digital/power-bi

Bitec. (s.f.). *Qlik Sense.* Obtenido de https://www.bitec.es/business-intelligence/qlik-sense/

Castillo Montaño, J. E. (1 de Mayo de 2000). *Beneficios de la Utilización de la Información, Basada en un Data Warehouse, Diseñada para un Departamento de una Universidad Privada -Edición Única.* Obtenido de https://repositorio.tec.mx/handle/11285/569541

Castro Jara, B. R. (Marzo de 2022). *Diseñar e implementar la metodología Hefesto para un Data Warehouse y Data Mining en un sistema ERP.* Obtenido de https://dspace.ups.edu.ec/handle/123456789/22684

Díaz García, E., & Cabrera Granado, E. (2018). *Manual de uso de Jupyter Notebook para aplicaciones docentes.* Obtenido de https://eprints.ucm.es/id/eprint/48304/

Forero Castañeda, D. A., & Sánchez García, J. A. (2022). Introducción a la inteligencia de negocios basada en la metodología KIMBALL. TIA Tecnología, investigación y academia. Obtenido de https://revistas.udistrital.edu.co/index.php/tia/article/view/18082

García Cortez, C. (2022). *DATA WAREHOUSE COMO PARADIGMA DE EFICIENCIA EN UNA EMPRESA.* Obtenido de http://ri.uaemex.mx/handle/20.500.11799/113275

García Merino, E. M., & García Merino, M. J. (2018). *Análisis de los Modelos de Inteligencia de Negocios basados en Big Data en las Pymes del Ecuador.* Obtenido de https://cienciaytecnologia.uteg.edu.ec/revista/index.php/cienciaytecnologia/article/view/157

García Pérez, A. M. (2020). Aplicación de técnicas de inteligencia de negocios y análisis de datos en el entorno empresarial cubano: retos y perspectivas. Revista Cubana de Ciencias Informáticas. Obtenido de http://scielo.sld.cu/scielo.php?pid=S2227-18992020000400191&script=sci_arttext

Garcia, M. J., & Rodas Silva, J. (2022). Análisis comparativo de metodologías y herramientas tecnológicas para procesos de Business Intelligence orientado a la toma de decisiones. Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones.

Gartner. (5 de Abril de 2023). *Magic Quadrant for Analytics and Business Intelligence Platforms.* Obtenido de https://www.gartner.com/doc/4247699

Gonzales López, R. A. (5 de Octubre de 2012). *Impacto de la Data Warehouse e Inteligencia de Negocios en el Desempeño de las Empresas: Investigación Empírica en Perú, como País en Vías de Desarrollo.* Obtenido de https://www.tdx.cat/handle/10803/85876#page=1

Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: the definitive guide to dimensional modeling.* Indianapolis: John Wiley & Sons.

Mendoza, A. (23 de Noviembre de 2022). *Metodologías Data Warehouse.* Obtenido de https://gravitar.biz/datawarehouse/metodologias-data-warehouse/

Microsoft. (30 de Julio de 2022). *Create and use basic area charts.* Obtenido de https://learn.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-basic-area-chart

Moffa, L. (2015). *Data warehouse e inteligencia de negocios aplicados a organismos del Estado.* Obtenido de https://repositoriocyt.unlam.edu.ar/handle/123456789/852

Naeem, T. (28 de Abril de 2020). *Qué es una herramienta ETL: tipos, características y casos de uso.* Obtenido de https://www.astera.com/es/knowledge-center/what-is-etl-tool/

Neteris. (s.f.). *Tableau Software - Una herramienta de visualización de datos interactiva.* Obtenido de https://info.neteris.com/tableau-software-visualizacion-datos/

Oracle. (s.f.). *¿Qué es un almacén de datos?* Obtenido de https://www.oracle.com/ar/database/what-is-a-data-warehouse/

Sánchez Espinoza, J. C., & Canelo Sotelo, C. A. (11 de Junio de 2019). *MODELO DE DATA WAREHOUSE CON APLICACION DE INTELIGENCIA DE NEGOCIOS PARA LAS PYMES.* Obtenido de http://www.revistas.unjbg.edu.pe/index.php/cyd/article/view/737

Tableau. (s.f.). *Bar Charts Understanding and using Bar Charts.* Obtenido de https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/bar-charts

Tableau. (s.f.). *Elija el tipo de gráfico adecuado para sus datos.* Obtenido de https://help.tableau.com/current/pro/desktop/es-es/what_chart_example.htm

Tableau. (s.f.). *Line Charts Understanding and using Line Charts.* Obtenido de https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/line-charts

Tableau. (s.f.). *Pie Charts Understanding and using Pie Charts.* Obtenido de https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/pie-charts

Tableau. (s.f.). *What Is Data Visualization? Definition, Examples, And Learning Resources.* Obtenido de https://www.tableau.com/learn/articles/data-visualization

Tecnologías Información. (s.f.). *Data Warehouse: Tipos, Aquitectura y Características.* Obtenido de https://www.tecnologias-informacion.com/datawarehouse.html#

Treviño Reyes, R., Rivera Rodríguez, F. S., & Garza Alonso, J. A. (2020). La analítica de datos como ventaja competitiva en las organizaciones. VINCULATEGICA EFAN. Obtenido de http://www.web.facpya.uanl.mx/Vinculategica/Vinculategica6_2/5_Trevi%C3%B1o_Rivera_Garza.pdf

UNIR. (29 de Noviembre de 2019). *Lenguaje R, ¿qué es y por qué es tan usado en Big Data?* Obtenido de https://www.unir.net/ingenieria/revista/lenguaje-r-big-data/

Zas Vázquez, A. A. (2022). *Construcción y explotación de un Data Warehouse para el análisis de información de una empresa de telecomunicaciones.* Obtenido de https://ruc.udc.es/dspace/handle/2183/32029

ANEXOS

Anexo A. Código de lenguaje Python empleado en la limpieza y depuración de los datos.

Importamos pandas

Generamos una ruta para que lea los archivos .csv

Generamos una lista con las variables que presentan en común todos los archivos

Procesamos y generamos un data frame principal

Imprimimos el total de filas (registros), el cual es igual a 87423

In [ ]:

```
import pandas as pd
import glob
import numpy as np

ruta_folder = '/content'
lista_archivos = glob.glob(ruta_folder + "/*.csv")
colnames=['Trans #','Type','Entered/Last Modified','Last modified by','Date','Num','Sour
ce Name','Name Address','Name Street1','Name Street2','Name City','Name State','Name
Zip','Name Contact','Name Phone #','Name Fax #','Name E-Mail','Name Account #','Me
mo','Adj','P. O. #','Name','Ship Date','Deliv Date','FOB','Via','Terms','Due Date','Billed D
ate','Paid Through','Ruc:','Email:','F.Pago:','Vendedor:','Ciudad:','Provincia:','Limite Cred
ito','Documento Garantia','Tipo ID','Contabilidad','Factura N/C','Fecha Fact N/C','Item','I
tem Description','SSN/Tax ID','Payroll Item','Income Subject To Tax','Wage Base','Wage
Base (Tips)','Account','Class','Rep','Sales Tax Code','Clr','Estimate Active','Billing Status
','Split','Print','Paid','Pay Meth','Aging','Open Balance','Qty','U/M','Sales Price','Debit','Cr
edit','Amount','Balance','Tax Table Version','User Edit?','Calculated Amount','Amount Di
fference','S. O. #','Account Type','Tax Line','WC Rate','Exp. Mod.','WC Code','State','Ac
tion','Peso','% Importacion','Backordered','Avg Days to Pay','Paid Date','Ship To City','S
hip To Address 1','Ship To Address 2','Ship To State','Ship Zip','Pay Period Begin Date','
Pay Period End Date','Check #','Other 1','Other 2','Preferred Delivery Method','Paycheck
Date','Current Rate','Previous Rate','% Change','Pay Period','Notes','Amount Paid','Last
Name','First Name','Contribution Amount','External Payroll ID']
df_principal = pd.DataFrame(pd.read_csv(lista_archivos[0],usecols=colnames))


for i in range(1,len(lista_archivos)):
    data = pd.read_csv(lista_archivos[i],usecols=colnames)
    df = pd.DataFrame(data)
    df_principal = pd.concat([df_principal,df],axis=0)

df_principal.shape[0]
```

Verificamos las primeras 5 filas con HEAD

In [ ]:

```
df_principal.head()
```

Generamos una copia para tener un backup de la información

In [ ]:

```
cpdf1=df_principal.copy()
```

```
cpdf1.head()
```

Verificación del tipo de columna

In [ ]:

```
for c in df_principal.columns:
   print(c, "TIPO-->",df_principal[c].dtype)
```

Verificación de variables con NULL

Almacenamiento en lista

In [ ]:

```
columnasnull=list()
for c in df_principal.columns:
  if(df_principal[c].isnull().sum()==df_principal.shape[0]):
    print("yes", df_principal[c].isnull().sum() , "columna "+c)
    columnasnull.append(c)

  else:
    print("no")
```

```
columnasnull
```

Visualización de lista con columnas NULL

In [ ]:

```
columnasnull
```

Eliminación de columnas NULL

In [ ]:

```
df_principal=df_principal.drop(columnasnull, axis=1)
df_principal
```

Todas las columnas de tipo OBJECT se transforman a Mayúsculas, además se eliminan espacio al inicio y al final de cada dato

In [ ]:

```
for c in df_principal.columns:
  if(df_principal[c].dtype==object):
    df_principal[c] = df_principal[c].astype(str)
    df_principal[c] = df_principal[c].str.upper()
    df_principal[c] = df_principal[c].str.rstrip()
    df_principal[c] = df_principal[c].str.lstrip()


df_principal
```

Cambio de object a float/int

Agrupamos

- Object - Se realiza agrupación para desplegar valores únicos
- Float - Se realiza un describe para ver valores

In [ ]:

```
for c in df_principal.columns:
  if(df_principal[c].dtype==object):
    print('(----------------------------------)')
    print(c)
    print( df_principal[c].unique())
    print('(----------------------------------)')
  elif(df_principal[c].dtype==float):
    print('(----------------------------------)')
    print(c)
    print(df_principal[c].describe())
    print('(----------------------------------)')
```

Se eliminan columnas de tipo float que no tienen valor, ya que todos sus datos son n/a o cero (0)

In [ ]:

```
columnas_cero=['Contribution Amount','Amount Paid','Current Rate','Avg Days to Pay']

df_principal=df_principal.drop(columnas_cero, axis=1)
df_principal
```

Transformación de object a numéricas, se analiza columna por columna

0) Columna Trans #

In [ ]:

```
columna_eliminada=[]
columna_analisis=df_principal.columns[0]
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce(",", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].str.replace(",", ".").asty
pe(int)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis].dtypes
```

2) Columna Entered/Last Modified

In [ ]:

```
columna_analisis=df_principal.columns[2]
df_principal[columna_analisis]=pd.to_datetime(df_principal[columna_analisis],format=
'%m/%d/%Y %H:%M:%S')
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

3) Columna Last modified by

In [ ]:

```
columna_analisis=df_principal.columns[3]
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.replace(" [DELETED USER]", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].replace("JANETH DIAZ", "SARITA DIAZ")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("SARITA", "SARITA DIAZ")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("LILI", "LILI QUINTANA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("GEOCO ESPIN", "GEOCONDA ESPIN")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("GEOCONDA", "GEOCONDA ESPIN")
df_principal[columna_analisis].unique()
```

4) Columna Date

In [ ]:

```
columna_analisis=df_principal.columns[4]
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: str(x.rsplit('/', maxsplit=1)[0])+str("/")+str(20)+x.rsplit('/', maxsplit=1)[-1] if len(x)<10 else x)
df_principal[columna_analisis]=pd.to_datetime(df_principal[columna_analisis],format='%m/%d/%Y')
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

5) Columna num

In [ ]:

```
columna_analisis=df_principal.columns[5]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","0")
df_principal[columna_analisis].fillna('0',inplace=True)
df_principal[columna_analisis]=test01=df_principal[columna_analisis].str.replace("001-001-","").astype(int)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

6) Columna Source Name

In [ ]:

```
columna_analisis=df_principal.columns[6]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("**SAITEL","SAITEL")

lista_elementos_unicos_ordenados=sorted(df_principal[columna_analisis].unique().tolist())

for elemento in lista_elementos_unicos_ordenados:
    print(elemento)
```

7) Columna Name Address

In [ ]:

```
columna_analisis=df_principal.columns[7]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","NING
```

```
UNO")
df_principal[columna_analisis].fillna("NINGUNO",inplace=True)
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
listaColumn7=df_principal[columna_analisis].unique().tolist()
sorted(listaColumn7)
```

8) Columna Name Street1

In [ ]:

```
columna_analisis=df_principal.columns[8]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","NING
UNO")
df_principal[columna_analisis].fillna("NINGUNO",inplace=True)
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
listaColumn8=df_principal[columna_analisis].unique().tolist()
sorted(listaColumn8)
```

9) Columna Name Street2

In [ ]:

```
columna_analisis=df_principal.columns[9]
df_principal[columna_analisis].unique()
#La columna se elimina por no ser relevante
columna_eliminada.append(columna_analisis)
```

10) Columna Name City

In [ ]:

```
columna_analisis=df_principal.columns[10]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
#La columna se elimina por no ser relevante
columna_eliminada.append(columna_analisis)
```

11) Columna Name State

In [ ]:

```
columna_analisis=df_principal.columns[11]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","NING
UNO")
df_principal[columna_analisis].fillna("NINGUNO",inplace=True)
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
listaColumn11=df_principal[columna_analisis].unique().tolist()
sorted(listaColumn11)
```

12) Columna Name Zip

In [ ]:

```
columna_analisis=df_principal.columns[12]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
#La columna se elimina por no ser relevante
columna_eliminada.append(columna_analisis)
```

13) Columna Name Contact

In [ ]:

```
columna_analisis=df_principal.columns[13]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","NING
UNO")
df_principal[columna_analisis].fillna("NINGUNO",inplace=True)
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
listaColumn13=df_principal[columna_analisis].unique().tolist()
sorted(listaColumn13)
```

14) Columna Name Phone #

In [ ]:

```
columna_analisis=df_principal.columns[14]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","0")
df_principal[columna_analisis].fillna("0", inplace = True)
df_principal[columna_analisis].unique().tolist()
#Los campos null se llenan con 0
```

15) Columna Name Fax #

In [ ]:

```
columna_analisis=df_principal.columns[15]
df_principal[columna_analisis].unique().tolist()
df_principal[columna_analisis].isnull().sum()
columna_eliminada.append(columna_analisis)
#cantidad elevada de valores null: 85703 de una data total de 87423
#Se elimina la columna
```

16) Columna Name E-Mail

In [ ]:

```
columna_analisis=df_principal.columns[16]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","")
listaColumn16=df_principal[columna_analisis].unique().tolist()
sorted(listaColumn16)
```

17) Columna Name Account #

In [ ]:

```
columna_analisis=df_principal.columns[17]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
#se considera en eliminar
columna_eliminada.append(columna_analisis)
```

18) Columna Memo

In [ ]:

```
columna_analisis=df_principal.columns[18]
df_principal[columna_analisis].unique().tolist()
#Se concidera eliminar por no ser relevante
```

```
columna_eliminada.append(columna_analisis)
columna_eliminada
```

19) Columna Name

In [ ]:

```
columna_analisis=df_principal.columns[19]
df_principal[columna_analisis].unique().tolist()
listaColumn19=df_principal[columna_analisis].unique().tolist()
sorted(listaColumn19)
```

20) Columna Ship Date

In [ ]:

```
columna_analisis=df_principal.columns[20]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","01/01/
1900")
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: str(x.rs
plit('/', maxsplit=1)[0])+str("/")+str(20)+x.rsplit('/', maxsplit=1)[-1] if len(x)<10 else x)
df_principal[columna_analisis]=pd.to_datetime(df_principal[columna_analisis],format=
'%m/%d/%Y')
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

21) Columna Terms

In [ ]:

```
columna_analisis=df_principal.columns[21]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis].unique().tolist()
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce("CRÉDITO", "CREDITO"))
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce("DÍAS", "DIAS"))
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["CASH","CO
NTADO POR DESPACHA"], "CONTADO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["2 DE ABRIL
2015","nan","NAN"], "NINGUNO")
sorted(df_principal[columna_analisis].unique().tolist())
```

22) Columna Due Date

In [ ]:

```
columna_analisis=df_principal.columns[22]
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: str(x.rs
plit('/', maxsplit=1)[0])+str("/")+str(20)+x.rsplit('/', maxsplit=1)[-1] if len(x)<10 else x)
df_principal[columna_analisis]=pd.to_datetime(df_principal[columna_analisis],format=
'%m/%d/%Y')
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

23) Columna Ruc:

In [ ]:

```
columna_analisis=df_principal.columns[23]
df_principal[columna_analisis].unique().tolist()
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","0")
df_principal[columna_analisis].fillna("0", inplace = True)
#datos nulos se coloca 0
```

24) Columna Email:

In [ ]:

```
columna_analisis=df_principal.columns[24]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","NING
UNO")
df_principal[columna_analisis].fillna("NINGUNO", inplace = True)
df_principal[columna_analisis].unique().tolist()
```

25) Columna F.Pago:

In [ ]:

```
columna_analisis=df_principal.columns[25]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce("CREDITO", "CRÉDITO"))
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce("DIAS", "DÍAS"))
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce("CHQ", "CHEQUE"))

df_principal[columna_analisis]=df_principal[columna_analisis].replace(["TRAN","TR
ANS","TRNS"], "TRANSFERENCIA")


df_principal[columna_analisis]=df_principal[columna_analisis].replace(["17927011390
01","EFRAINLLAUTONG1977@GMAIL.COM","GE"], "")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["CANTADO",
"CIONTADO","CO NTADO","COMD","CONT","CONTA","CONTADOO","COTAD
O","0"], "CONTADO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace({"15 DÍAS":"C
RÉDITO 15 DÍAS","21 DÍAS":"CRÉDITO 21 DÍAS","30 DÍAS":"CRÉDITO 30 DÍAS
"})
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["CHE","CHE
Q","CH"], "CHEQUE")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("CH 30 DÍAS",
"CHEQUE 30 DÍAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["EFEC","EEC
","EFEC - TRANS","EFECT"], "EFECTIVO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["PAYPAL","P
YPALL"], "PAY PAL")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("GARANTIA",
"GARANTIAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("PAYCLUB", "
PAY CLUB")
```

```
df_principal[columna_analisis]=df_principal[columna_analisis].replace('TARJETA DE
BITO  ','TARJETA DEBITO')


df_principal[columna_analisis].unique().tolist()
listaColumn25=df_principal[columna_analisis].unique().tolist()
sorted(listaColumn25)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')


#por la variabilidad y por la similitud con TERMS se concidera eliminar la columna
columna_eliminada.append(columna_analisis)
columna_eliminada
```

26) Columna Vendedor:

In [ ]:

```
columna_analisis=df_principal.columns[26]
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["CTADO","C
ONTADO"],"OF")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["OF -","OF -
NUEVO","OF- NUEVO","OF-NUEVO","INT","INTERNED"],"OF")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("GES","GE")
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","NA")
df_principal[columna_analisis].fillna("NA",inplace = True)
```

27) Columna Ciudad:

In [ ]:

```
columna_analisis=df_principal.columns[27]
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["AMABATO",
"ANBATO"],"AMBATO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["ASOGUEZ",
"AZOGUEZ"],"AZOGUES")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["BABAHOYA
","BABAHO"],"BABAHOYO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["BAHÍA DE C
ARÁQUEZ","BAHÍA DE CARAQUEZ"],"BAHIA DE CARAQUEZ")
df_principal[columna_analisis]=df_principal[columna_analisis].replace([", STO.DOMI
NGO","SANTO  DOMINGO","SANTO DOMINGO","SANTO DOMINGO DE LOS
COLORADOS","SANTO DOMNIGO","STO. DE LOS TSACHILAS","STO. DOMIN
GO","STO.DOMINGO"],"SANTO DOMINGO DE LOS TSACHILAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("BLAZAR","B
ALZAR")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["CAÑAR","C
A?AR"],"CANAR")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("CENTINELA
DEL CONDOR /","CENTINELA DEL CONDOR")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("ELGUABO","
EL GUABO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("ESMERLAD
AS","ESMERALDAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["FCO. DE OR
```

```
ELLANA","SAN FRANCISCO","SAN FRANCISCO DE ORELLANA"],"FRANCISC
O DE ORELLANA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["GUAYAQIIL
","GUAYAQIL","GUAYQUIL"],"GUAYAQUIL")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["JOYA DE LO
S SACHA","JOYA DE LOS SACHAS","LA JOYA DE LOS SACCHAS","LA JOYA D
E LOS SACHA"],"LA JOYA DE LOS SACHAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("LAGO  AGRI
O","LAGO AGRIO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["LOGRO?O",
"LOGROÑO"],"LOGRONO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("LOLA","LOJ
A")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("MACRA","M
ACARA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("MOMTECRIS
TI","MONTECRISTI")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("MORONA","
MORONA SANTIAGO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["NAJANJAL",
"NARNJAL"],"NARANJAL")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("MACAHALA
","MACHALA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("OTVALO","O
TAVALO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("PEDRO VICE
NTE  MALDONADO","PEDRO VICENTE MALDONADO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("PELILIEO","
PELILEO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("PIMANPIRO"
,"PIMAMPIRO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("POROTVIEJ
O","PORTOVIEJO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("QUINIDE","Q
UININDE")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["QUTIO","QU
ITO3"],"QUITO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("RIOBAMAB
A","RIOBAMBA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["RUMI?AHUI
","RUMIÑAHUI"],"RUMINAHUI")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["SAN GABRE
L","SAN GABRIL"],"SAN GABRIEL")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["SAN  MIGU
EL  BANCOS","SAN MIGUEL DE LOS  BANCOS"],"SAN MIGUEL DE LOS BAN
COS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("SANTA  ELE
NA","SANTA ELENA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("SAQUICILI",
"SAQUISILI")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("SHUSHUFIN
DO","SHUSHUFINDI")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("STA. ROSA",
"SANTA ROSA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("TEMA","TEN
```

```
A")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("TULCÁN","T
ULCAN")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("URDANTE","
URDANETA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","NING
UNA")
df_principal[columna_analisis].fillna("NINGUNA",inplace = True)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis].isnull().sum()
#se remplaza los null por NINGUNA
```

28) Columna Provincia:

In [ ]:

```
columna_analisis=df_principal.columns[28]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)


df_principal[columna_analisis]=df_principal[columna_analisis].replace(["0","GE","nan
","NINGUNO","NAN"],"NINGUNA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("ST KITTS","E
XTERIOR")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["INBABURA"
,"IBARRA","IMABURA"],"IMBABURA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["LA JOYA DE
LOS SACHAS","FCO- DE ORELLLANA","FCO. DE ORELLANA","FRANCISCO D
E ORELLANA"],"ORELLANA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace([", STO.DOMI
NGO DE LOS TS","SANTO DOMINGO","SANTO DOMINGO DE LOS TSACHILA
","SANTO DOMINGO D","SANTO DOMINGO DE LOS TSA","SANTO DOMINGO
DE LOS TSÁCHILAS","SANTO DOMINGO TSACHILAS","SANTO DOMINSGO D
E LOS TSACHILA","STO DOMINGO TSACHILAS","STO. DOMINGO","STO. DO
MINGO DE LOS TS","STO. DOMINGO DE LOS TSACHILAS","SANTO DOMING
O DE LOS  TSACHILA","SANTO DOMINGO  DE LOS TSACHILA"],"SANTO DO
MINGO DE LOS TSACHILAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["ARCHIDON
A","TENA"],"NAPO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["ASOGUEZ",
"AZOGUES","AZOGUEZ","CAÑAR"],"CANAR")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["BOL?VAR","
BOLÍVAR","GUARANDA"],"BOLIVAR")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["CHINBORA
ZO","CHMBORAZO","RIOBAMBA"],"CHIMBORAZO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["ESMARALD
AS","ESMERLADAS"],"ESMERALDAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("GALÁPAGOS
","GALAPAGOS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("GUAYAQUIL
","GUAYAS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["LOS RÍOS","
LOS R?OS","LO RIOS","QUINSALOMA"],"LOS RIOS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["MANAB?","
MANABÍ","MANANBI","NANABI"],"MANABI")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("LOJO","LOJA
```

```
")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["MACHALA"
,"SANTA ROSA"],"EL ORO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["MORONA S
ANTIAGO","MORONONA SANTIAGO"],"MORONA SANTIAGO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["PICHICNHA
","PICHIINCHA","PICHINCHIA","PICHINHCA","QUITO"],"PICHINCHA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["SUCUMB?O
S","SUCUMBÍOS","SHUSHUFINDI"],"SUCUMBIOS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["TUNGUARA
HUA","TUNGURAGUA","TUNGUTAHUA","TUGURAHUA"],"TUNGURAHUA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("PAZTAZA","
PASTAZA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("ZAMORA","
ZAMORA CHINCHIPE")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["LA LIBERTA
D","STA. ELENA"],"SANTA ELENA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("TULCAN","C
ARCHI")

df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
sorted(df_principal[columna_analisis].unique().tolist())
```

29) Columna Limite Credito

In [ ]:

```
columna_analisis=df_principal.columns[29]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)


df_principal[columna_analisis]=df_principal[columna_analisis].replace(["0 NINGUNO
","0 NINGUO","0.","CHIMBORAZO","CONTADO","CREDITO","EN BLANCO","E
SMERALDAS","GE","GUAYAS","LETRA","LOS RIOS","NIGUNO","NINGUNA","
NINGUNO","O","PICHINCHA","SANTA CRUZ","nan","NAN"],"0.0")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("2,000","2000"
)
df_principal[columna_analisis]=df_principal[columna_analisis].replace("3,500.00","35
00.00")
df_principal[columna_analisis]=df_principal[columna_analisis].astype(float).astype(int)

df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

30) Columna Documento Garantia

In [ ]:

```
columna_analisis=df_principal.columns[30]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)


df_principal[columna_analisis]=df_principal[columna_analisis].replace(["NAN","nan",
"0","CONTADO","ESMERLADAS","PICHINCHA","R","MINGUNO","N","N INGU
NO","N/G","NI NGUNO","NIGUNO","NIMGUNO","NINCHA","NINGNO","NINGN
UNO","NINGNUO","NINGUN","NINGUN O","NINGUN0","NINGUNI","NINGUNI
O","NINGUNO","NINGUNO+","NINGUNO|","NINGUNP","NINGUO","NNGUNO"]
,"NINGUNA")
```

```
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["LC","LETRA
","LETRA  DE CAMBIO","LETRA DECAMBIO","LETRA CAMBIO"],"LETRA DE
CAMBIO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["LETRA CA
MBIO EN BLANCO","LETRA DE CAMBIO EN BLANCO","LETRA  FIRMADA E
N BLANCO"],"LETRA EN BLANCO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("LETRA  2000
","LETRA 2000")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("LETRA 700.0
0","LETRA 700")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["PAGAR?","P
AGARÉ"],"PAGARE")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("3000","LETR
A 3000")
```

```
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

31) Columna Tipo ID

In [ ]:

```
columna_analisis=df_principal.columns[31]

df_principal[columna_analisis]=df_principal[columna_analisis].replace(["0","N","S","
NAN"],"NINGUNO")
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["PASAPORTE
","PSS"],"P")

df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis].fillna("NINGUNO",inplace = True)
#Los null se los cataloga como NINGUNO
```

32) Columna Contabilidad

In [ ]:

```
columna_analisis=df_principal.columns[32]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NO","N")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("SI","S")
df_principal[columna_analisis]=df_principal[columna_analisis].replace([".","1","A","R
","X","NAN"],"NINGUNO")
df_principal[columna_analisis].fillna("NINGUNO",inplace = True)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
#Los null se los cataloga como NINGUNO
```

33) Columna Factura N/C

In [ ]:

```
columna_analisis=df_principal.columns[33]

df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

34) Columna Fecha Fact N/C

In [ ]:

```
columna_analisis=df_principal.columns[34]

df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

35) Columna Item

In [ ]:

```
columna_analisis=df_principal.columns[35]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce("KIT0", "KIT:KIT0"))
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x if ':' i
n x else ':'+x)
familia=df_principal[columna_analisis].str.split(pat=":",n=1, expand=True)
familia=familia[0]
sku=df_principal[columna_analisis].str.rsplit(pat=":",n=1, expand=True)
sku=sku[1]
df_principal["ITEM_FAMILIA"]=familia
df_principal["ITEM_SKU"]=sku
df_principal
```

36) Columna Item Description

In [ ]:

```
columna_analisis=df_principal.columns[36]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","SIN D
ESCRIPCION")
df_principal[columna_analisis].fillna("SIN DESCRIPCION",inplace = True)
#Datos con null se coloca SIN DESCRIPCION
```

37) Columna Account

In [ ]:

```
columna_analisis=df_principal.columns[37]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
#columna con formato definido en la parte contable
```

38) Columna Rep

In [ ]:

```
columna_analisis=df_principal.columns[38]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

39) Columna Sales Tax Code

In [ ]:

```
columna_analisis=df_principal.columns[39]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)

df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

40) Columna Split

In [ ]:

```
columna_analisis=df_principal.columns[40]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

41) Columna Print

In [ ]:

```
columna_analisis=df_principal.columns[41]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

42) Columna Paid

In [ ]:

```
columna_analisis=df_principal.columns[42]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

43) Columna Aging

In [ ]:

```
columna_analisis=df_principal.columns[43]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

44) Columna Open Balance

In [ ]:

```
columna_analisis=df_principal.columns[44]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

45) Columna Qty

In [ ]:

```
columna_analisis=df_principal.columns[45]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.replace(",", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].replace("nan","0")
df_principal[columna_analisis]=df_principal[columna_analisis].astype(float).astype(int)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

46) Columna Sales Price

In [ ]:

```
columna_analisis=df_principal.columns[46]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.replace(",", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].astype(float)
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","0.00")
df_principal[columna_analisis].fillna("0.00",inplace=True)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

47) Columna Debit

In [ ]:

```
columna_analisis=df_principal.columns[47]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.replace(",", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].astype(float)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis].isnull().sum()
columna_eliminada.append(columna_analisis)
```

48) Columna Credit

In [ ]:

```
columna_analisis=df_principal.columns[48]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.replace(",", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].astype(float)
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","0.00")
df_principal[columna_analisis].fillna("0.00",inplace=True)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

49) Columna Amount

In [ ]:

```
columna_analisis=df_principal.columns[49]
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.replace(",", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].astype(float)
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","0.00")
```

```
df_principal[columna_analisis].fillna("0.00",inplace=True)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

50) Columna Balance

In [ ]:

```
columna_analisis=df_principal.columns[50]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

51) Columna User Edit?

In [ ]:

```
columna_analisis=df_principal.columns[51]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

52) Columna S. O. #

In [ ]:

```
columna_analisis=df_principal.columns[52]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis].describe()
df_principal[columna_analisis].isnull().sum()
columna_eliminada.append(columna_analisis)
```

53) Columna Account Type

In [ ]:

```
columna_analisis=df_principal.columns[53]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

54) Columna Tax Line

In [ ]:

```
columna_analisis=df_principal.columns[54]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

55) Columna State

In [ ]:

```
columna_analisis=df_principal.columns[55]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
```

56) Columna Action

In [ ]:

```
columna_analisis=df_principal.columns[56]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

57) Columna Backordered

In [ ]:

```
columna_analisis=df_principal.columns[57]
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","0")
df_principal[columna_analisis].fillna("0",inplace=True)
df_principal[columna_analisis]=df_principal[columna_analisis].astype(str)
df_principal[columna_analisis]=df_principal[columna_analisis].apply(lambda x: x.repla
ce(",", ""))
df_principal[columna_analisis]=df_principal[columna_analisis].astype(float).astype(int)
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

58) Columna Ship To City

In [ ]:

```
columna_analisis=df_principal.columns[58]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
columna_eliminada
```

59) Columna Ship To Address 1

In [ ]:

```
columna_analisis=df_principal.columns[59]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
df_principal[columna_analisis]=df_principal[columna_analisis].replace("NAN","SIN D
IRECCION")
df_principal[columna_analisis].fillna("SIN DIRECCION",inplace=True)
```

60) Columna Ship To Address 2

In [ ]:

```
columna_analisis=df_principal.columns[60]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
columna_eliminada
```

61) Columna Ship To State

In [ ]:

```
columna_analisis=df_principal.columns[61]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
columna_eliminada
```

62) Columna Ship Zip

In [ ]:

```
columna_analisis=df_principal.columns[62]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

```
columna_eliminada.append(columna_analisis)
columna_eliminada
```

63) Columna Preferred Delivery Method

In [ ]:

```
columna_analisis=df_principal.columns[63]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
columna_eliminada.append(columna_analisis)
columna_eliminada
```

64) Columna ITEM_FAMILIA

In [ ]:

```
columna_analisis=df_principal.columns[64]
df_principal[columna_analisis]=df_principal[columna_analisis].replace(["FIBRA -ACC
ESORIOS","ACCESORIOS DE FIBRA","BOBINAS DE FIBRA"],"FIBRA OPTICA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("VARIOSITE
MS","VARIOS ITEMS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("DEMOS PRO
YECTOS","EQUIPOS DEMOS PROYECTOS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("EQUIPO VID
EO VIGILANCIA","EQUIPOS VIDEO VIGILANCIA")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("ACSESORIO
S","ACCESORIOS")
df_principal[columna_analisis]=df_principal[columna_analisis].replace("Z.EQ.CURSO
S CERTIFICACION","CURSOS DE CERTIFICACION")

df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')

lista_elementos_unicos_ordenados=sorted(df_principal[columna_analisis].unique().tolis
t())
for elemento in lista_elementos_unicos_ordenados:
    print(elemento)
```

65) Columna ITEM_SKU

In [ ]:

```
columna_analisis=df_principal.columns[65]
df_principal.groupby([columna_analisis]).size().reset_index(name='cantidad')
```

Creación de columnas de latitud y longitud para provincia

In [ ]:

```
cod_provincias = [{
        "nombre": "Cuenca",
        "provincia": "AZUAY",
        "longitud": -2.9005500,
        "latitud": -79.0045300,
        "habitantes": "712.127"
        }, {
        "nombre": "Guaranda",
        "provincia": "BOLIVAR",
```

"longitud": -1.5926300,
"latitud": -79.0009800,
"habitantes": "183.641"

}, {
"nombre": "Azogues",
"provincia": "CANAR",
"longitud": -2.7396900,
"latitud": -78.8486000,
"habitantes": "225.184"
}, {
"nombre": "Tulcán",
"provincia": "CARCHI",
"longitud": 0.8118700,
"latitud": -77.7172700,
"habitantes": "164.524"
}, {
"nombre": "Riobamba",
"provincia": "CHIMBORAZO",
"longitud": -1.6709800,
"latitud": -78.6471200,
"habitantes": "458.581"
}, {
"nombre": "Latacunga",
"provincia": "COTOPAXI",
"longitud": -0.9352100,
"latitud": -78.6155400,
"habitantes": "409.205"
}, {
"nombre": "Machala",
"provincia": "EL ORO",
"longitud": -3.2586100,
"latitud": -79.9605300,
"habitantes": "600.659"

}, {
"nombre": "Esmeraldas",
"provincia": "ESMERALDAS",
"longitud": 0.9592000,
"latitud": -79.6539700,
"habitantes": "534.092"

}, {
"nombre": "Puerto Baquerizo Moreno",
"provincia": "GALAPAGOS",
"longitud": -0.62881,
"latitud": -90.36387,
"habitantes": "25.124"
}, {
"nombre": "Guayaquil",
"provincia": "GUAYAS",
"longitud": -2.2058400,
"latitud": -79.9079500,
"habitantes": "3.645 483"

}, {

"nombre": "Ibarra",
"provincia": "IMBABURA",
"longitud": 0.3517100,
"latitud": -78.1223300,
"habitantes": "398.244"
}, {
"nombre": "Loja",
"provincia": "LOJA",
"longitud": -3.9931300,
"latitud": -79.2042200,
"habitantes": "448.966"
}, {
"nombre": "Babahoyo",
"provincia": "LOS RIOS",
"longitud": -1.8021700,
"latitud": -79.5344300,
"habitantes": "778.115"
}, {
"nombre": "Portoviejo",
"provincia": "MANABI",
"longitud": -1.0545800,
"latitud": -80.4544500,
"habitantes": "1.369.780"
}, {
"nombre": "Macas",
"provincia": "MORONA SANTIAGO",
"longitud": -2.3086800,
"latitud": -78.1113500,
"habitantes": "147.940"
}, {
"nombre": "Tena",
"provincia": "NAPO",
"longitud": -0.9938000,
"latitud": -77.8128600,
"habitantes": "103.697"
}, {
"nombre": "Francisco de Orellana",
"provincia": "ORELLANA",
"longitud": -0.933333,
"latitud": -75.666667,
"habitantes": "136.396"
}, {
"nombre": "Puyo",
"provincia": "PASTAZA",
"longitud": -1.4836900,
"latitud": -78.0025700,
"habitantes": "83.933"

}, {
"nombre": "Quito",
"provincia": "PICHINCHA",
"longitud": -0.2298500,
"latitud": -78.5249500,
"habitantes": "2.576.287"
}, {

```
    "nombre": "Santa Elena",
    "provincia": "SANTA ELENA",
    "longitud": -2.2262200,
    "latitud": -80.8587300,
    "habitantes": "308.693"

    }, {
    "nombre": "Santo Domingo",
    "provincia": "SANTO DOMINGO DE LOS TSACHILAS",
    "longitud": -0.2530500,
    "latitud": -79.1753600,
    "habitantes": "368.013"

    }, {
    "nombre": "Nueva Loja",
    "provincia": "SUCUMBIOS",
    "longitud": -0.083333,
    "latitud": -76.883333,
    "habitantes": "176.472"
    }, {
    "nombre": "Ambato",
    "provincia": "TUNGURAHUA",
    "longitud": -1.2490800,
    "latitud": -78.6167500,
    "habitantes": "504.583"

    }, {
    "nombre": "Zamora",
    "provincia": "ZAMORA CHINCHIPE",
    "longitud": -4.040649,
    "latitud": -78.948716,
    "habitantes": "91.376"

    }, {
    "nombre": "N/A",
    "provincia": "NINGUNA",
    "longitud": 0.0000000,
    "latitud": 0.0000000,
    "habitantes": "N/A"

    }, {
    "nombre": "N/A",
    "provincia": "EXTERIOR",
    "longitud": 0.0000000,
    "latitud": 0.0000000,
    "habitantes": "N/A"

    }

  ]


columna_analisis=df_principal.columns[28]
longitud=list()
latitud=list()
```

```
for dato_provincia in df_principal[columna_analisis]:
    for num_provincia in  range(len(cod_provincias)):
        if dato_provincia in cod_provincias[num_provincia].get('provincia'):
            longitud.append(cod_provincias[num_provincia].get('latitud')),
            latitud.append(cod_provincias[num_provincia].get('longitud'))


df_principal["LONGITUD"]=longitud
df_principal["LATITUD"]=latitud
```

Borrar datos del df que están en la lista columna_eliminada

In [ ]:

```
df_principal.drop(columna_eliminada, axis=1, inplace=True)
```

Se revisa si alguna columna tiene valores núlos

In [ ]:

```
newdf=df_principal.isnull().sum()
df = pd.DataFrame(newdf)
df
```

Exportación FINAL

In [ ]:

```
df_principal.to_csv('Final01.csv')
```