



UNIVERSIDAD NACIONAL DE ENTRE RÍOS

FACULTAD DE INGENIERÍA

Tecnicatura Universitaria en Procesamiento y Explotación de Datos

Materia: Espacio Integrador 2

Nombre de la Actividad: Informe final

**Título: "Automatización del análisis de ofertas y optimización del
manejo de stock y ventas en LAR durante el 2024"**

Profesor: Lic. Walter R. Elias

Alumno: Gareis, Daiana Micaela



Índice

Introducción.....	3
Contexto del Informe.....	3
Situación Problemática.....	4
Objetivos generales.....	5
Objetivos específicos.....	5
Hipótesis.....	6
Marco Teórico.....	7
Materiales y métodos.....	10
Resultados.....	20
Conclusión.....	27
Proyección.....	27
Bibliografía.....	28
Anexos.....	30

Introducción

Contexto del Informe

En el marco de la asignatura '**Espacio Integrador 2**' de la carrera **Tecnicatura Universitaria en Procesamiento y Explotación de Datos**, dictada en la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos, se elaboró este informe con el propósito de documentar el análisis realizado sobre un problema presentado por la empresa La Agrícola Regional Cooperativa Limitada (LAR) y registrar las propuestas de solución basadas en datos. LAR es una organización con una fuerte presencia en la provincia de Entre Ríos, que se dedica a la comercialización de productos en sus sucursales mayoristas y minoristas, donde las promociones tienen influencia en el comportamiento de los consumidores.

En 2024, se enfrenta al desafío de automatizar el análisis de las ofertas promocionales lanzadas en sus sucursales, con el objetivo de mejorar la eficiencia en la evaluación del impacto que estas tienen sobre las ventas y el stock de productos. La empresa ha solicitado el desarrollo de una solución tecnológica que permita procesar automáticamente la información y generar reportes estadísticos que faciliten la toma de decisiones.

Este proyecto busca desarrollar una herramienta automatizada que no solo optimice el tiempo dedicado a la recolección y procesamiento de datos, sino que también ofrezca a LAR la posibilidad de ajustar sus estrategias de oferta basándose en datos cuantitativos y medibles. La automatización del proceso permitirá identificar patrones en las ventas, evaluar la efectividad de las promociones en distintas sucursales y ajustar la gestión del inventario en función del comportamiento de la demanda.

Situación Problemática

El problema que enfrenta la empresa puede definirse de la siguiente manera: se encuentran ante el desafío de automatizar el análisis y la presentación de resultados de las ofertas promocionales realizadas en las sucursales mayoristas y minoristas de LAR en Entre Ríos durante el año 2024. Actualmente, debido a la ausencia de automatización, resulta complicado realizar una evaluación eficaz del impacto de las promociones en las ventas y en el stock de productos.

La situación se profundiza debido a la necesidad de procesar grandes volúmenes de información de manera periódica, con diversas sucursales que presentan comportamientos de venta distintos, y que manejan productos de diferentes tipos. Esta carencia de una herramienta automatizada limita a la empresa en su capacidad para obtener una visión clara y en tiempo real sobre la efectividad de sus promociones, afectando directamente su habilidad para ajustar sus estrategias de marketing y gestionar el inventario de manera óptima.

Por lo tanto, resulta fundamental desarrollar una solución que permita normalizar y organizar de manera eficiente los datos de las ofertas, integrando esta información con el historial de ventas y el manejo del stock de productos. Asimismo, esta herramienta debe facilitar la evaluación del impacto de las promociones, tanto en términos de aumento de ventas como en el porcentaje de stock vendido, proporcionando reportes que sean accesibles y claros para apoyar la toma de decisiones estratégicas dentro de la empresa.

Con esto surge, **¿Cómo se puede automatizar el análisis de ofertas y presentación de resultados en LAR para maximizar la eficiencia del proceso y mejorar el manejo del stock y las ventas en sus sucursales durante el mes de Agosto del 2024?**

Objetivos generales

El objetivo principal de este proyecto es automatizar el proceso de análisis de las ofertas promocionales realizadas por LAR en sus sucursales mayoristas y minoristas de Entre Ríos durante el mes de Agosto durante el año 2024. La finalidad de esta automatización es evaluar el impacto de las ofertas en términos de aumento de ventas, porcentaje de stock vendido y la efectividad en la atracción de clientes.

Objetivos específicos

- Desarrollar scripts automatizados para normalizar la lista de productos en oferta, asegurando la coherencia y organización de los datos necesarios para su análisis.
- Integrar la información de las ofertas normalizadas con el historial de ventas y datos de stock, permitiendo identificar patrones de comportamiento en las promociones realizadas.
- Calcular indicadores clave como el aumento en las ventas, el porcentaje de stock vendido durante las promociones y el efecto en la atracción de clientes para cada sucursal.
- Diseñar reportes claros y visualmente comprensibles que presenten los resultados obtenidos, brindando a la empresa una visión integral de la efectividad de sus estrategias promocionales.

Hipótesis

Dentro del contexto descrito previamente, se puede plantear la siguiente hipótesis:

Si se implementa una herramienta automatizada para el análisis de las ofertas promocionales de LAR, se espera que estas ofertas tengan un impacto significativo tanto en el aumento de la cantidad de ventas como en el porcentaje de stock vendido de los productos ofertados. Al automatizar el procesamiento de datos y la comparación de las promociones con el historial de ventas y el manejo del inventario, la empresa podrá identificar de manera precisa cómo las ofertas influyen en el comportamiento de compra de los clientes, lo que le permitirá ajustar sus estrategias comerciales con mayor efectividad.

Esta hipótesis parte de la suposición de que las promociones bien estructuradas y correctamente monitoreadas a través de una herramienta automatizada generarán un incremento en la demanda de los productos ofertados, logrando que un alto porcentaje del stock disponible sea vendido durante el período promocional. Además, se espera que este sistema permita a LAR ajustar sus campañas promocionales en función de los resultados obtenidos, optimizando así tanto las ventas como la gestión de inventarios.

Esta suposición será evaluada y validada a lo largo del desarrollo del proyecto, permitiendo analizar si efectivamente las ofertas generan los impactos esperados sobre las ventas y el porcentaje de stock vendido de los productos.

Marco Teórico

La Agrícola Regional Cooperativa Limitada, Agropecuaria, de Consumo y de Servicios Públicos es una empresa de propiedad conjunta democráticamente controlada por más de 5.000 asociados, representados por un Consejo de Administración, y administrada por una gerencia general que se sustenta en un grupo de gestión y el trabajo de más de medio millar de empleados. Fundada en 1910, ha desempeñado un rol central en el desarrollo productivo de la región, consolidándose como una herramienta clave para el crecimiento económico local. (La Agrícola Regional Cooperativa Limitada, 2024)

Las ofertas semanales de LAR abarcan una amplia variedad de productos, desde alimentos básicos y productos de consumo masivo hasta insumos agropecuarios y artículos para el hogar, diseñados para satisfacer las necesidades tanto de familias como de productores rurales. Este enfoque diverso, sumado a sus más de 100 años de historia, ha permitido a la cooperativa adaptarse a las demandas del mercado y mantener una posición relevante en el sector. (La Agrícola Regional Cooperativa Limitada, 2024) [1]

La Agrícola Regional es una empresa que cuenta con un área especializada en el análisis de datos. Actualmente, almacena y procesa su información mediante un enfoque basado en agrupación en clúster multidimensional (MDC). Este método proporciona una solución eficiente para organizar datos en tablas a lo largo de varias dimensiones de manera flexible, continua y automática. El MDC permite a la empresa gestionar información clave relacionada con los artículos del supermercado, las ofertas, las secciones, los grupos, las sucursales, entre otros aspectos relevantes.” (LAR, 2024) [2].

Para abordar la problemática planteada, me proporcionaron tablas específicas del sistema de LAR, incluyendo: **artículos**, **listas_ofertas**, y la familia de tablas **fac_descuentos**, entre otras. Como parte del trabajo, realicé la normalización de la tabla de ofertas, cuyo resultado será integrado al sistema manejado por la empresa a través de Apache NiFi, garantizando una incorporación eficiente y automatizada de los datos procesados.

A lo largo del informe, menciono términos específicos al ámbito en que se desarrolla el trabajo que es importante aclarar:

- **Sucursal:** Según la Real Academia Española (RAE), una sucursal es un "establecimiento que se encuentra en un lugar diferente al de la central de la que depende, pero que realiza las mismas funciones". (RAE, 2024) [3]

- **Descuento:** Según la Escuela de Negocios CEUPE, un descuento es una acción que permite reducir el precio de un producto o servicio, generalmente en porcentajes, como estrategia para incentivar su compra o contratación. [4]
- **Oferta:** Según la RAE, una oferta se define como "el conjunto de bienes o mercancías que se presentan en el mercado con un precio concreto y en un momento determinado". (RAE, 2024) [5]. Una oferta es una estrategia de venta que puede ser un descuento, pero también se puede presentar una oferta para un periodo del año específico, hasta agotar el stock de un producto, etc. Aunque oferta y descuento suelen usarse indistintamente, en este contexto hacen referencia a conceptos distintos.

Para abordar la problemática planteada, la empresa adicionalmente, me proporcionó diversas fuentes de datos para el análisis:

1. **Precios y costos unificados (desde el 30-06):** Este conjunto de datos incluye información sobre los precios y costos de los productos desde junio hasta septiembre de 2024. Estos datos fueron facilitados por el departamento de compras de LAR.
2. **Existencia Agosto:** Este archivo CSV contiene información sobre la cantidad de productos disponibles durante el mes de agosto de 2024, también suministrado por el departamento de compras.
3. **Ventas realizadas en agosto de 2024:** La empresa compartió diferentes archivos CSV referidos a las ventas realizadas durante ese mes:
 - **Ventas Super Agosto**
 - **Ventas Carnicería Agosto 2024**
 - **Ventas Mayorista Agosto 2024**

Estos archivos de ventas, a diferencia de los anteriores, proceden de fuentes externas y no forman parte de la base de datos multidimensional ni son generados por el departamento de compras.

A lo largo de mi carrera, cursé diversas materias que me aportaron conocimientos fundamentales para llevar a cabo este trabajo. En primer lugar, destaco la materia que tuvo mayor relevancia en el desarrollo del mismo, seguida de aquellas que contribuyeron al desarrollo de habilidades iniciales también valiosas.

- **Base de datos:** la estructura de las tablas del trabajo, corresponde con una base de datos relacional. Esto se debe a que "en una base de datos relacional las tablas

representan entidades (por ejemplo, artículos, ventas, listas_oferta) y las columnas dentro de las tablas son los atributos de esas entidades”. Silberschatz, A. (2007) [6] Aunque las relaciones entre las tablas no están directamente visibles en el Diagrama de Entidad-Relación (DER), pueden inferirse mediante el uso de columnas comunes como *código*, *barras* o *codigo_interno*, un aspecto característico de las bases de datos relacionales.

Además, en esta asignatura adquirí conocimientos sobre cómo realizar consultas, tanto básicas como complejas, lo cual me sirvió para relacionar las tablas y extraer la información necesaria para este trabajo.

- Otras materias son Comprensión Lectora y Producción Escrita, Informática Básica, Espacio Integrador I y Algoritmos y Estructura de Datos.

La primera fue fundamental para aprender a redactar textos propios de manera adecuada y citar correctamente fuentes y trabajos realizados por terceros. Por su parte, Informática Básica y Espacio Integrador I me proporcionaron una base sólida para adquirir las primeras herramientas de programación, así como para familiarizarme con el uso de Python y las bibliotecas más importantes de este lenguaje. Por último, Algoritmos y Estructura de Datos fue fundamental para profundizar mis conocimientos en Python, particularmente en la creación y manejo de funciones, módulos y objetos.

Finalmente, los conocimientos adquiridos en *Exploración de Datos Multivariados* también fueron de gran utilidad, ya que esta asignatura me permitió gestionar bases de datos más grandes y, mediante una correcta limpieza y normalización de los datos, obtener los resultados esperados de manera eficiente.

La herramienta utilizada para integrar lo desarrollado durante la práctica con el sistema implementado en la empresa fue *Apache NiFi*. Esta plataforma, es “un *sistema distribuido dedicado a extraer, transformar y cargar datos (ETL)*. Es Open Source y está desarrollado y mantenido por la Apache Software Foundation”. Fernandez, O. (2019, febrero 24). [7]

Materiales y métodos

Forma de desarrollo de la práctica

Desarrollé la práctica de forma virtual, manteniendo reuniones semanales de aproximadamente 30 minutos todos los jueves a través de Microsoft Teams. Utilizamos una hoja de Excel para registrar y llevar el control de las horas realizadas. Para consultas, explicaciones o notificaciones sobre cambios en las tareas, nos comunicamos por correo electrónico a través de Gmail, y para el intercambio de archivos empleamos Google Drive como medio principal.

Además, tuvimos un encuentro presencial que duró aproximadamente cuatro horas, durante el cual comenzamos a integrar lo que había desarrollado con Apache NiFi y realizamos un control más detallado de los avances logrados hasta ese momento.

Herramientas Generales

Trabajé con dos herramientas principales para obtener los artículos en oferta: **PostgreSQL**, utilizando **PgAdmin4**, y **Python**. Desde PgAdmin4 realicé el cruce de las tablas necesarias, mientras que con Python generé las consultas SQL para obtener los mismos resultados a partir de queries y, mediante la librería *pandas*, creé el archivo CSV con los artículos en oferta.

Posteriormente, para cruzar las ofertas con las ventas y realizar su análisis, opté por usar Python a través de **Visual Studio Code**, ya que esta herramienta resultó ser más eficiente para automatizar el proceso requerido, el cual debía integrarse mediante **Apache NiFi**. Por último, utilicé **Metabase** para generar gráficos y también para cargar las tablas a la base de datos que no logré cargar directamente desde PgAdmin.

Las herramientas utilizadas se destacan por sus capacidades:

- **Python**, descrito como un lenguaje de programación potente y fácil de aprender, proporciona estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. [8]
- **PgAdmin4** es la plataforma de desarrollo y administración de código abierto más popular y con más cantidad de funciones para PostgreSQL considerada por muchos como una de las bases de datos de código abierto más avanzada del mundo. [9]

- **PostgreSQL** es un sistema de gestión de bases de datos relacional de objetos (ORDBMS) de código abierto con una sólida reputación de fiabilidad y rendimiento. Según su documentación, sus funciones principales son almacenar y recuperar datos de forma segura y, al mismo tiempo, dar soporte a las mejores prácticas informáticas actuales. [10]

Obtención artículos oferta

Para crear las tablas y llenar la base de datos comencé con los DDL iniciales compartidos por la empresa y con los archivos csv para poder poblar las mismas. Las tablas que efectivamente terminé usando para el análisis fueron: **articulos**, **listas_oferta**, **fac_descuentos**, **fac_descuentos_conceptos**, **fac_descuentos_detalle_aplicacion**. Estas tablas contienen información que detallan las ofertas y descuentos aplicados durante el año junto con otras tablas, que contienen información sobre otros aspectos de estos artículos que estuvieron en oferta.

Para poder acceder desde python a la base de datos que generé en PgAdmin y poder operar con las tablas de la misma, tuve que usar la biblioteca psycopg2 que le permite a Python utilizar una base de datos PostgreSQL.

El desafío del trabajo consistió en poder unificar los datos de la totalidad de los productos que se encontraron en oferta, y para ello lo que se debía hacer es juntar la familia de tablas **fac_descuentos** con la tabla **listas_oferta** que eran tablas muy diferentes, y poder generar una nueva fuente de datos que contenga qué productos estuvieron en oferta únicamente en el mes de agosto, durante qué rango de fechas, el porcentaje de descuento aplicado a cada producto y cuál era el stock que había al inicio en cada sucursal en cada periodo que el producto estuvo en oferta. Inicialmente sólo analicé el mes de agosto, porque el csv de Existencia sólo contiene información de este mes y porque de este mes tenía todos los datos disponibles.

Lo que primero hice es definir una CTE ya que me permite definir una consulta temporal y el resultado que esta produce lo puedo referenciar en la consulta principal. Al

principio, decidí definir el porcentaje de descuento en 0 y luego calcularlo. Luego determino el rango de fechas que voy a considerar, para que corresponda al mes de agosto.

Una vez obtenido el producto y el periodo de fechas que estuvo en oferta, crucé estos artículos en oferta con dos datasets históricos correspondientes a los precios de los artículos y la existencia de estos artículos durante el mes de agosto. Para ello realizo la creación de tablas en la base de datos: **Precios_y_costos** que llené con el csv **Precios y costo_unificado desde 30-06** y **stock_sucursal** usando los datos del csv **Existencia_Agosto**.

El primer csv tenía los costos y precios minoristas y mayoristas de todos los productos vendidos por la empresa a partir del mes de junio, a este lo usé para tomar el precio minorista de los productos. Lo que hice para obtener el porcentaje de descuento aplicado al producto, en cada sucursal y en cada periodo de oferta es calcular la diferencia entre el precio que figura en **Precios_y_costo** tomando una fecha justo antes del inicio de la oferta y el precio en oferta obtenido de la tabla **listas_oferta**. A este resultado lo multiplico por 100 y lo redondeo a dos decimales usando la función `round()`.

Luego utilizo la función CASE de PostgreSQL que me permite realizar evaluaciones condicionales y devolver diferentes valores basados en las condiciones especificadas. En este caso, si el precio en la tabla listas ofertas o el precio minoristas en **Precios_y_costo** era NULL, el porcentaje de descuento se determinaba como nulo, en caso contrario la función calcula el porcentaje de descuento.

El segundo csv contenía la existencia de todos los productos comercializados por la empresa durante ese mismo mes, y lo que busco obtener es el stock al inicio de la oferta, por eso tomo el stock un día antes del inicio de la misma. También utilizo la función CASE, porque tengo dos opciones a considerar: en el caso que la sucursal es NULL se asigna una cadena vacía y si la existencia es NULL, se asigna 0.

Luego de ejecutar el query para la obtención de los artículos en oferta, ejecuto otro para obtener la cantidad de artículos diferentes que se ofertan. Esto lo realice porque un mismo producto puede estar ofertado durante diferentes periodos y en diferentes resultados. Es por lo mismo que la cantidad de registros es un número mayor que la cantidad de artículos en oferta.

En un primer momento, quise utilizar python para cargar la base de datos y cruzar la misma con los csv de **Existencia** y **Precios_y_costo** . Luego opté simplemente por ejecutar el query realizado desde PostgreSQL en el archivo llamado **Obtencion_articulos_oferta.py** ya que era más fácil gestionar y cruzar las tablas que se habían generado por este medio desde python, que realizar todo el cruce únicamente con código python.

Utilizo un LEFT JOIN para incluir todas las filas de la tabla de ofertas. Este me permite realizar cálculos y comparaciones incluso cuando ciertas piezas de datos no están disponibles, devolviendo NULL en esos casos en lugar de excluir la fila completa de los resultados.

Durante el intento de normalización de los artículos en oferta decidimos realizar un cambio referido a quitar del requerimiento de solución al análisis de las ofertas provenientes de la familia de tablas **fac_descuentos**. Esto se decidió de esta manera debido al tiempo que teníamos y en pos de intentar alcanzar un resultado valioso y fácilmente verificable con el tiempo disponible con los datos de la tabla **listas_oferta**.

Para cargar este archivo en PgAdmin para posteriormente cruzarlo con el csv **Ventas_Combinadas_Agosto** se tuvo que hacer un proceso de normalización de variables ya que en **Articulos_oferta** y en **stock_sucursal** los nombres de sucursales aparecían como 'CRESPO-CCC', 'HERNANDEZ-SMINO', etc. entonces opté por cambiarlas a 'Crespo', 'Hernandez'. Este cambio de nombres lo hago a través de un **mapeo de valores** para estandarizar los nombres de las sucursales en el DataFrame **stock_sucursal** y

Artículos_Oferta. Para realizar esto, también utilizo un diccionario llamado **cambios_nombres**, que especifica cómo transformo los valores originales de la columna **sucursal** a los nombres que deseo usar.

Ambas formas de escribir las variables son correctas según snake case para nombres de variables en Python. Ambas cumplen con lo que esta dice referido a los nombres de las variables que deben estar en minúsculas, con las palabras separadas por guiones bajos según sea necesario para mejorar la legibilidad [11]. Elegí la opción de los nombres más cortos y directos ya que eran más fáciles de leer y entender.

El archivo CSV correspondiente a los artículos en oferta fue generado utilizando la librería *pandas* y se guardó con el nombre **Articulos_oferta**. Posteriormente, se cargó en PgAdmin, creando una tabla llamada **articulos_of**.

Obtención de ventas en oferta

Para obtener las ventas relacionadas con las ofertas, opté por trabajar en ambos lenguajes, Python y SQL. Inicialmente, preferí SQL, ya que me resultaba más intuitivo analizar los resultados directamente desde las tablas en PgAdmin. Sin embargo, también realicé el análisis en paralelo utilizando Python, siguiendo una solicitud específica de la empresa, y pude confirmar que los resultados obtenidos en ambos enfoques coincidían. El archivo donde ejecuto el query realizado por PostgreSQL lo llamo **Archivo_Ventas.py** y el que uso para realizar todo el análisis de las ventas en python lo llamo **Ventas_Python.py**

Los csv referidos a las ventas se combinaron usando la función `merge()` en python para obtener un dataset de ventas unificado que se llamó **Ventas_Combinadas_Agosto**. Al mismo lo cargue para realizar el cruzamiento con la tabla **Articulos_oferta** que ya habia subido previamente y así poder obtener cuantas unidades durante el periodo de oferta se vendió de los artículos ofertados.

Lo que busco lograr es obtener cuantos productos se venden en una sucursal y en un periodo específico, y esto se obtiene agrupando por los campos **sucursal** y **fecha_inicio** y **fecha_fin**. Se utiliza un INNER JOIN que incluye únicamente las filas donde hay correspondencia entre la tabla de ofertas y de ventas.

Además, realice tres gráficos usando diferentes librerías que proporciona python. Utilizo pyplot que es un módulo de la librería **matplotlib**. También uso **seaborn** que es una librería independiente que se basa en **matplotlib**, que proporciona gráficos más atractivos y fáciles de interpretar.

El primer gráfico que realizo es un gráfico de barras y lo hago para mostrar cómo varían las ventas a principios, mediados y fin del mes de agosto. Para ello hago una función **clasificar_periodo_fecha** que recibe una fecha como parámetro y asigna “Inicio del mes” a las fechas del 1-10 de agosto, “Mitad del mes” a las fechas del 10-20 y “Fin de mes” a las fechas del 20-30 de agosto.

El segundo gráfico lo realizo por medio de Metabase y me permite mostrar la totalidad de la tasa de ventas en oferta durante el mes de agosto en las cuatro sucursales consideradas: Crespo, Hernandez, Nogoyá Sur y Nogoyá Italia. El tercer gráfico lo realizó desde Python usando **Matplotlib (plt)** y **Seaborn (sns)**. Este muestra los 5 productos en oferta más vendidos durante la oferta.

Para cargar el archivo **Ventas_Combinadas_Agosto** a PgAdmin4 tuve que realizar también un proceso de limpieza de datos, comenzando con el formato de las fechas que estaban en formato excel y con la función de pandas *pd.to_datetime()* las transformo a formato datetime. Luego reviso si el dataset tiene valores nulos, y no eliminé datos porque no estaban en variables importantes. Por último, transformé la variable **total_ventas_unidad** a entero ya que este dato no podía tener decimales.

Obtención de resultados de Ventas en oferta

Al archivo generado de **ventas_oferta** lo quise subir a PgAdmin haciendo una limpieza previamente con python. El problema es que estaba teniendo un error que no pude solucionar. Sin embargo, al mismo lo logro subir a la base de datos a partir de Metabase y a la tabla la nombro como **resultado_ventas_oferta**. Luego de obtener las ventas realizadas de cada periodo del producto en oferta en cada una de las sucursales, procedí a realizar diferentes análisis de las ventas en oferta.

- Aumento de ventas respecto al promedio de ventas en días de comercialización normal. Para calcular esto, lo que hice es utilizar python y el archivo llamado **Resumen_Resultados.py** y partiendo del csv de **ventas_combinadas_agosto** y el csv de **resultado_ventas_oferta** generados anteriormente, realice una función llamada **calcular_ventas_normales** que calcula las ventas normales y el aumento de ventas para un producto en una sucursal y periodo dados. La idea es considerar la misma cantidad de días para el periodo de oferta y periodo de venta normal.

El problema que estaba teniendo es que no siempre existe una cantidad equivalente de días en oferta y de días normales. Además existen productos ofertados que estuvieron en oferta varias veces durante el mes de agosto y en algunos casos por más de 10 días. De esta manera no es posible conseguir un periodo equivalente de días normales. Por lo mismo, lo que hago es definir un umbral de 5 días para hacer la comparación. Si hay menos días, tomo esta cantidad de días disponibles.

El resultado se muestra en un csv generado por python llamado **comparacion_ventas.csv** que contiene las mismas columnas que el csv **resultado_ventas_ofertas** pero con tres columnas adicionales: **cantidad_ventas_normal**, que hace referencia al total de las ventas de un producto durante un periodo que no estuvo en oferta en una sucursal específica, **fechas_normal**, que contiene las fechas consideradas en el periodo que el producto no estuvo ofertado, **aumento_ventas**, que determina el porcentaje de aumento de ventas en los días de oferta en comparación con un periodo que el producto no

estuvo en oferta, si es positivo, significa que se vendió más durante la oferta, si es negativo significa que el producto se vendió en mayor cantidad en días que no estuvo ofertado y, por último, **cantidad_dias** que indica la cantidad de días que se usa en la comparación de los días de oferta y días sin oferta.

La fórmula que utilizo para calcularla es la siguiente:

$$\text{aumento_ventas} = \left(\frac{\text{Total Ventas en la Oferta} - \text{Total Ventas Normales}}{\text{Total Ventas Normales}} \right) \times 100$$

En el código también realizo una verificación de Ventas Combinadas que consiste en tomar el código de un producto, la fecha que se vendió y la sucursal y sumar el Total de Ventas por Unidad. A este resultado lo comparo con el resultado que obtengo en `comparacion_ventas`.

También realizo dos gráficos a través de python. El primero es un gráfico de filas, que es una variante del gráfico de barras. Lo que muestra es los productos clasificados en cuatro categorías de aumento de ventas: Aumento Mayor a 0, Aumento menor a 0, Sin dato, Aumento = 0. Para mostrar estos resultados en este informe, opte por utilizar una tabla que obtengo por medio de Google Sheet, ya que considero que se muestran mejor los resultados con la misma.

Por otro lado, también hago otro gráfico de filas para mostrar el aumento de ventas que se registra en cada sucursal por medio de python.

- Para calcular el porcentaje del stock disponible que se vendió en la oferta lo que hago es utilizar python y crear el archivo **Resumen_Resultados2.py** y partir del csv **resultado_ventas_oferta** y para cada producto en cada periodo que estuvo en oferta en cada sucursal, dividir el total de las ventas por el stock disponible y a esto multiplicarlo por 100 y redondearlo a dos decimales. Para guardar el resultado genero el csv **resultado_stock_vendido** que contiene una columna adicional

llamada **Porcentaje_Vendido**. La fórmula que usó para calcularla es la siguiente que me la proporcionó la empresa:

$$Porcentaje_Vendido = \left(\frac{Total_Ventas_Oferta}{Stock_Disponible} \right) \times 100$$

También realizó un gráfico de filas por Metabase que muestra el porcentaje del stock que se vendió en cada sucursal. Para ello realizó un query que luego también agrego en python que utiliza la fórmula para calcular el porcentaje vendido agrupando por sucursal.

- Por último, calculo la efectividad de la oferta por día también en el archivo **Resumen_Resultados2**. Lo que obtengo con ello es la cantidad de productos que en promedio se vendieron por día durante la oferta en una sucursal y periodo de oferta específico. Para guardar los resultados genero un csv llamado **resultado_efectividad_por_dia** que contiene los mismos campos que el csv **resultado_ventas_oferta** con 2 columnas adicionales **Dias_Oferta** y **Efectividad_Ventas_por_Dia**. La fórmula usada para calcular la efectividad de las ventas por día es la siguiente:

$$Efectividad_Ventas_por_Dia = \frac{Total\ Ventas\ Unidad\ (sumadas\ en\ el\ periodo)}{Días\ Oferta\ (sumados\ en\ el\ periodo)}$$

Integración con Apache Nifi

Los resultados obtenidos de los artículos en oferta se comenzaron a integrar durante el encuentro presencial con el sistema que tiene la empresa. Lo que se hizo durante la reunión presencial fue ver en general el entorno de Apache Nifi y cargar por el momento la tabla listas ofertas general. Lo que falta es adaptar la consulta de artículos oferta que tenía

los nombres de variables que yo use a los nombres que ya están en el sistema. Realizar esto que falta, supera el tiempo destinado a desarrollar la práctica.

Resultados

La cantidad de registros obtenidos en el csv **Artículos_oferta** fueron 1665 y obtuve un total de 8 columnas estas son: **'codigo'**, **'barras'**, **'descripcion'**, **'fecha_inicio'**, **'fecha_fin'**, **'porcentaje_descuento'**, **'sucursal'**, **'stock'**.

El total de artículos es: 371. La razón de la diferencia entre la cantidad de registros y de artículos es que un mismo artículo puede estar en oferta en diferentes periodos y ofertarse en 4 sucursales diferentes. Al quitar los resultados que estaban en la familia de tablas **fac_descuentos**, pase de tener 1681 registros a 1665, es decir que no se redujo de manera significativa la cantidad de registros.

A algunos registros no les calculé el **porcentaje_descuento** debido a que no estaba el precio en el csv de **Precios y costo_unificado desde 30-06** o no estaba en la columna **precios** de la tabla **listas_oferta**. Si bien eran casos muy puntuales, no podía realizar el cálculo y en ese caso al **porcentaje_descuento** le coloqué null. Por otro lado, en el caso del stock, hay algunos registros que figuran con stock negativo. En este caso no coloque null sino que los deje como aparecían porque estos mismos valores eran los que figuraban en el sistema.

Al realizar el cruce del csv de **Articulos_oferta** y **Ventas_combinadas_Agosto** en python y de las tablas **resultado_ventas_oferta** y **articulos_of** en PgAdmin4 obtuve un total de 1197 registros y 8 columnas que son las siguientes: **'codigo'**, **'Descripcion_Articulo_Oferta'**, **'Marca_producto'**, **'sucursal'**, **'fecha_inicio'**, **'fecha_fin'**, **'stock'**, **'Total_Ventas_Unidad'** que corresponden al total de artículos vendidos de todos los productos en oferta y este resultado se obtiene agrupando los registros por sucursales y los diferentes periodos en que el producto se ofertó.

El gráfico que se muestra a continuación muestra cómo varían las ventas durante el mes de agosto. Como se puede observar en el gráfico al principio del mes es cuando hay más ventas. A mediados y fin de mes no se observan diferencias significativas en las mismas. Esto puede deberse a que al principio del mes es cuando las personas más compran y que las ventas suelen disminuir a fin de mes.



Gráfico N°1 (Fuente: Elaboración propia)

El gráfico muestra la tasa de ventas en oferta por sucursal, donde la sucursal Nogoyá Italia tiene la mayor tasa, con más del 5% (0.05) de las ventas totales atribuidas a productos en oferta. Esto quiere decir que por cada 100 productos vendidos en la sucursal, 5 pertenecían a la categoría de productos en oferta. En contraste, la sucursal Crespo presenta la menor tasa, con un poco más del 2% (0.02) de las ventas totales correspondientes a productos en oferta. Esto refleja una mayor proporción de ventas en oferta en Nogoyá Italia comparada con las otras sucursales.

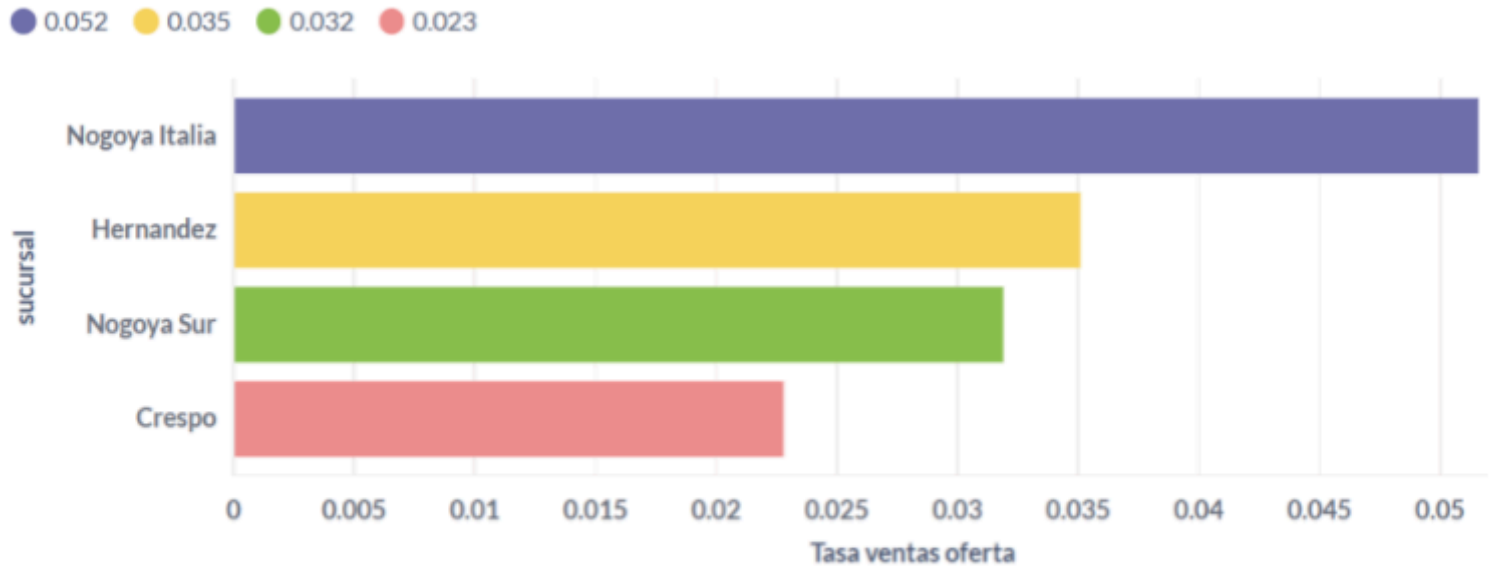


Gráfico N°2 (Fuente: Elaboración propia)

El siguiente gráfico de barras muestra los 5 productos más vendidos durante el periodo de oferta.

Top 5 Productos Más Vendidos

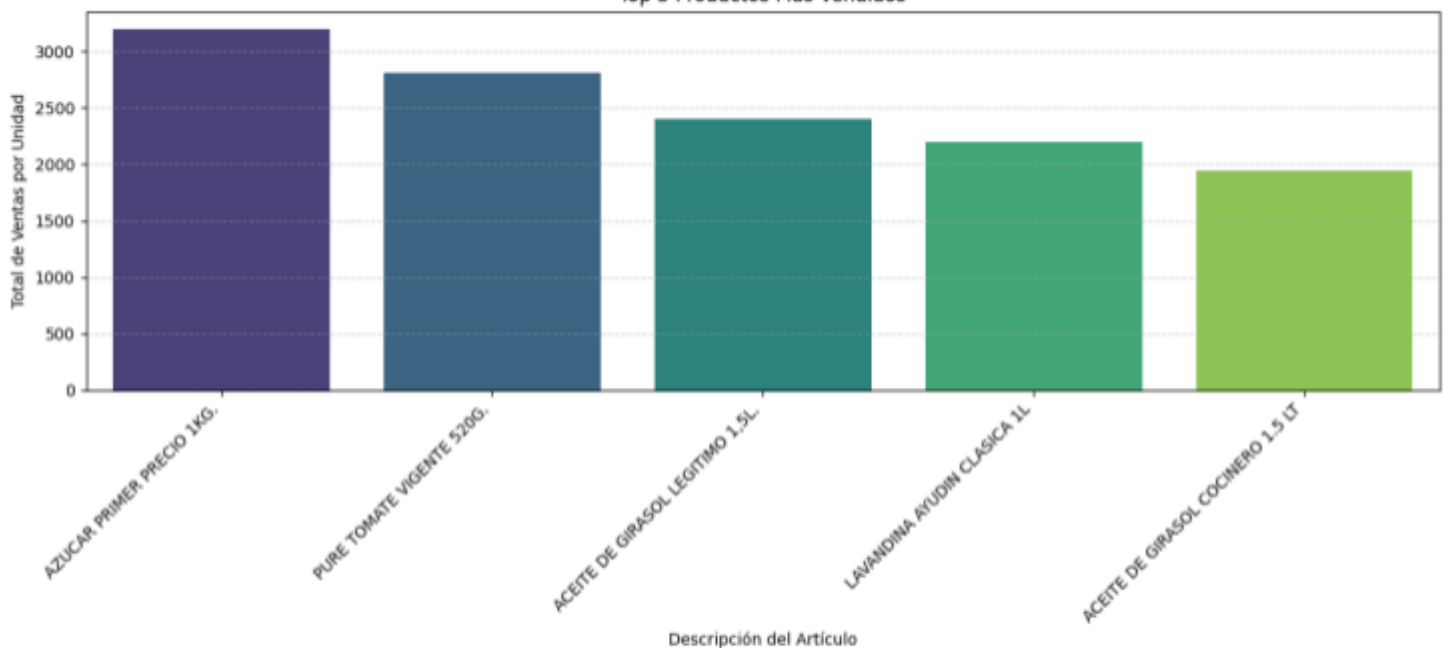


Gráfico N°3 (Fuente: Elaboración propia)

Al calcular el aumento de ventas respecto a promedio de ventas de días de comercialización normal, obtuve los resultados que se muestran en la siguiente tabla:

	Cantidad registros	Porcentaje
Aumento>0	703	58.78%
Aumento<0	384	32.11%
Sin Datos	48	4.01%
Aumento=0	61	5.10%
	1196	100%

Tabla N°1 (Fuente: Elaboración propia)

Como se observa en la tabla, hay un total de 58.78% productos que tuvieron un aumento en las ventas debido a las ofertas. Además también obtuve que en un 32.11% de los casos, en los días normales las personas compran más que en los días de oferta. En este análisis es importante también tener en cuenta que hay algunos productos que estuvieron en oferta por solo un día, y que el día tomado que no estuvo en oferta se seleccionó de forma aleatoria. En este sentido, la comparación no es tan específica ya que pudo darse el caso que ese producto en cualquier otro día sin oferta se venda en menor cantidad.

En el siguiente gráfico de barras se muestra como fue el resultado del aumento de ventas según las sucursales. Este gráfico muestra la cantidad de ventas en oferta y ventas en periodos normales en términos absolutos. El gráfico muestra en el eje horizontal (x) los porcentajes del aumento promedio de ventas. Como se puede observar, tanto en Crespo como en Nogoya Sur el aumento de las ventas fue superior a un 100%. Esto se debe a que en estas sucursales las ventas en tiempos de oferta aumentan de manera significativa en comparación con las ventas en periodos normales, lo que quiere decir que las ofertas incrementan significativamente la demanda de productos.

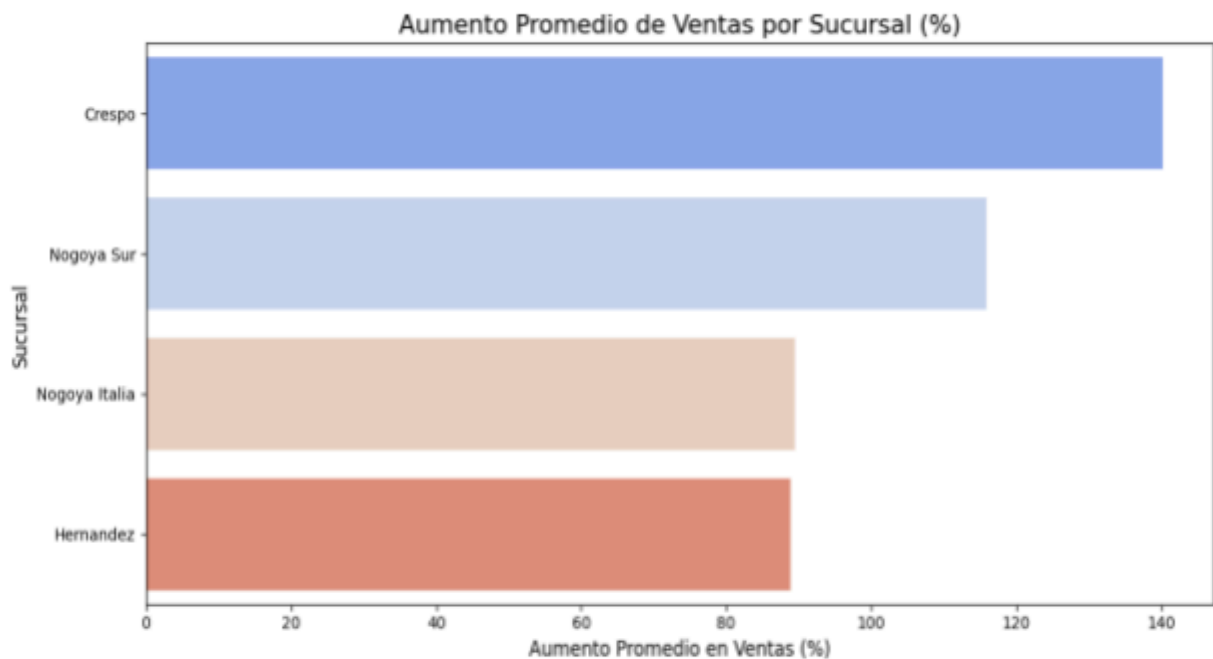


Gráfico N°4 (Fuente: Elaboración propia)

Al analizar el porcentaje del stock vendido en cada sucursal se puede observar en el siguiente gráfico de filas que la sucursal que más porcentaje vendió fue Crespo con una gran diferencia, seguida de Hernandez y Nogoyá Italia que tienen una mínima diferencia y en el último lugar se encuentra Nogoya Sur.



Gráfico N°5 (Fuente: Elaboración propia)

Conclusión

Analizar la parte de los datos que hace referencia a las ofertas realizadas durante el mes de agosto del año 2024 permite entender cómo estas afectan el interés de los consumidores por comprar los variados productos que se venden. Esto a su vez tiene un impacto directo en la cantidad de stock que queda del producto y como aumenta la necesidad de contar con una mayor cantidad de productos para poder satisfacer esta demanda de los productos ofertados.

Luego del análisis realizado se obtuvo que hay una diversa cantidad de productos que se ofertaron durante el mes, con variados porcentajes de descuento aplicados y que estuvieron en oferta en diferentes periodos y en la mayoría de los casos en las 4 sucursales analizadas. De todos ellos los 5 productos más vendidos en el mes fueron: azúcar, puré de tomate, aceite de girasol (legítimo) , lavandina, aceite de girasol (cocinero).

Al analizar las ventas en oferta por sucursal, se observa que Nogoyá Sur registró el mayor volumen de ventas en oferta, mientras que Crespo presentó el menor. Esto sugiere que las ofertas en Crespo no tuvieron un impacto significativo en aumentar la cantidad total de productos vendidos debido a los descuentos. Por el contrario, en las demás sucursales, la tasa de ventas en oferta respecto al total de ventas fue mayor, lo que indica que las promociones sí incentivaron a los consumidores a adquirir más productos por la reducción en el precio.

Si bien es cierto que el porcentaje de aumento de ventas <0 no es un número chico, ya que representa un 32.11% del total, puedo decir que, en general, las sucursales tuvieron un aumento de ventas considerablemente bueno. Los valores reflejados en el gráfico generado indican que todas las sucursales experimentaron un incremento significativo en promedio, siendo el aumento más bajo cercano al 90% (Hernández) y el más alto superando el 140% (Crespo). Esto sugiere que, en general, las estrategias aplicadas referidas a las ofertas lograron un impacto positivo en las ventas.

Con respecto al porcentaje del stock disponible que se vendió en la oferta, se observa que Crespo logró vender el 84.51% del total de su stock en oferta, superando significativamente a las demás sucursales, que no alcanzaron el 50%. Esto evidencia que, en términos de capacidad para vender productos en promoción, Crespo tuvo un mejor desempeño. Por lo tanto, considerando la hipótesis planteada, se puede concluir que el

porcentaje vendido de los productos ofertados en Crespo es alto, mientras que en las demás sucursales este indicador es de nivel medio-bajo.

Durante la obtención de los resultados pude obtener que Crespo registró el mayor volumen de ventas en oferta, lo cual es esperable considerando su mayor tamaño y cantidad de clientes. Sin embargo, la tasa de ventas en oferta respecto al total de ventas fue mayor en las otras sucursales, lo que indica que, proporcionalmente, las ofertas en las demás sucursales tuvieron un impacto más significativo en la decisión de compra de los consumidores.

El análisis desarrollado confirma la importancia de contar con una herramienta automatizada que facilite el procesamiento y análisis de grandes volúmenes de datos sobre las promociones. Las diferencias en el desempeño entre sucursales, especialmente el caso de la sucursal Crespo que tiene un comportamiento diferente a las otras, evidencian que las estrategias de promoción deben adaptarse a las características locales. Una solución automatizada permitiría normalizar los datos de manera eficiente, evaluar el impacto de las promociones en tiempo real y optimizar tanto las ventas como la gestión de stock, contribuyendo directamente a resolver la problemática identificada en LAR.

Proyección

Actualmente, el proceso de integración de los artículos en oferta con el sistema de gestión de la empresa está en curso, pero aún no se ha completado. Esto significa que, aunque se han dado los primeros pasos para automatizar la identificación y el análisis de los productos en oferta, el desarrollo de esta funcionalidad no está terminado y requiere de un proceso de transformación de datos por medio de Apache Nifi para su implementación completa.

Además, el cálculo de la elasticidad precio de la demanda para cada artículo ofertado quedó pendiente debido a que era necesario contar con un cálculo preciso del aumento en las ventas para poder obtener este indicador. Este análisis es fundamental para comprender cómo las variaciones en el precio afectan la demanda de los productos y, en consecuencia, permite evaluar de manera más precisa la efectividad de las promociones realizadas. Incorporar este indicador en el sistema será un objetivo clave para su desarrollo futuro.

Bibliografía

- *La Agrícola Regional Coop. Ltda.* (s/f). Lar.coop. Recuperado el 5 de diciembre de 2024, de <https://www.lar.coop/quienes>
- La Agrícola Regional, C. E. R. (Recuperado el 5 de diciembre de 2024)
- (S/f). Rae.es. Recuperado el 1 de diciembre de 2024, de <https://dle.rae.es/sucursal>
- (S/f). Rae.es. Recuperado el 1 de diciembre de 2024, de <https://dle.rae.es/oferta>
- *Descuento: Qué es, ejemplos y tipos.* (s/f). Ceupe. Recuperado el 1 de diciembre de 2024, de <https://www.ceupe.com/blog/descuento.html>
- Silberschatz, A. (2007). *Fundamentos de Bases de Datos*. McGraw-Hill Interamericana.
- Fernandez, O. (2019, febrero 24). *Apache NiFi: Introducción*. Aprender BIG DATA; AprenderBigData. <https://aprenderbigdata.com/introduccion-apache-nifi/>
- (s. f.). *Tutorial de Python (Versión 3.0)*. Recuperado el 4 de diciembre de 2024, de <https://docs.python.org/es/3/tutorial/>
- Page, D. (s/f). *PgAdmin - PostgreSQL Tools*. Pgadmin.org. Recuperado el 4 de diciembre de 2024, de <https://www.pgadmin.org/>
- *¿Qué es PostgreSQL?* (2022, enero 19). Purestorage.com; Pure Storage. <https://www.purestorage.com/es/knowledge/what-is-postgre-sql.html>
- *PEP 8 – style guide for Python code.* (s/f). Python Enhancement Proposals (PEPs). Recuperado el 1 de diciembre de 2024, de <https://peps.python.org/pep-0008/>

Anexos

```
CREATE TABLE "DBA"."listas_oferta" (
    "barras" CHAR(13) NOT NULL,
    "precio" DECIMAL(12,2) NULL,
    "operador" CHAR(15) NULL,
    "fecha_inicio" DATE NOT NULL,
    "fecha_fin" DATE NULL,
    "observaciones" CHAR(35) NULL,
    "fecha" DATE NULL DEFAULT CURRENT DATE,
    "hora" TIME NULL DEFAULT CURRENT TIME,
    "vigente" CHAR(1) NOT NULL,
    "clase" INTEGER NOT NULL,
    PRIMARY KEY ( "barras" ASC, "fecha_inicio" ASC, "vigente" ASC,
"clase" ASC )
) IN "system";
COMMENT ON COLUMN "DBA"."listas_oferta"."vigente" IS 'Novedad Activa';
```

```
CREATE TABLE "DBA"."fac_descuentos" (
    "codigo" INTEGER NOT NULL,
    "fecha_inicio" DATE NOT NULL,
    "fecha_fin" DATE NOT NULL,
    "porcentaje" NUMERIC(4,2) NOT NULL,
    "dias" INTEGER NOT NULL,
    "habilitada" CHAR(1) NOT NULL,
    "fecha_alta" DATE NOT NULL,
    "incluyendo_total_detalle" INTEGER NULL,
    "acumulativa" INTEGER NULL,
    PRIMARY KEY ( "codigo" ASC )
) IN "system";
```

```
CREATE TABLE "DBA"."fac_descuentos_conceptos" (
    "codigo" INTEGER NOT NULL,
    "descripcion" VARCHAR(50) NULL,
    "descri_corta" CHAR(5) NULL,
    PRIMARY KEY ( "codigo" ASC )
) IN "system";
```

```
CREATE TABLE "DBA"."fac_descuentos_detalle_aplicacion" (
    "codigo" INTEGER NOT NULL,
    "item" INTEGER NOT NULL,
    "seccion" NUMERIC(5,0) NULL,
    "departamento" NUMERIC(5,0) NULL,
    "grupo" NUMERIC(5,0) NULL,
    "marca" NUMERIC(5,0) NULL,
    "articulo" NUMERIC(6,0) NULL,
    "porcentaje" NUMERIC(5,2) NULL,
```

```

        PRIMARY KEY ( "codigo" ASC, "item" ASC )
    ) IN "system";

CREATE TABLE "DBA"."articulos" (
    "codigo" DECIMAL(6,0) NOT NULL DEFAULT 0,
    "barras" VARCHAR(13) NOT NULL,
    "enganchado" CHAR(1) NOT NULL DEFAULT '0',
    "descripcion" CHAR(50) NULL DEFAULT 'ARTICULO S/DESCRIPCION',
    "descripcion_corta" CHAR(35) NULL DEFAULT 'DESCRI CORTA',
    "margen_venta_porcentaje" DECIMAL(4,2) NOT NULL DEFAULT 0,
    "impuesto_interno_porcentaje" DECIMAL(4,2) NOT NULL DEFAULT 0,
    "habilitado_sn" CHAR(1) NOT NULL DEFAULT 'S',
    "modalidad_factura" CHAR(2) NOT NULL DEFAULT 'X',
    "imagen_articulo" CHAR(100) NOT NULL DEFAULT
    'c:\\Sistemas\\Graficos\\Productos',
    "unidades" DECIMAL(12,2) NOT NULL DEFAULT 0,
    "id_iva" DECIMAL(1,0) NOT NULL DEFAULT 1,
    "id_marca" DECIMAL(6,0) NOT NULL DEFAULT 0,
    "id_seccion" DECIMAL(2,0) NOT NULL DEFAULT 0,
    "id_departamento" DECIMAL(5,0) NOT NULL DEFAULT 0,
    "id_rubro" DECIMAL(4,0) NOT NULL DEFAULT 0,
    "id_grupo" NUMERIC(5,0) NOT NULL DEFAULT 0,
    PRIMARY KEY ( "barras" ASC )
) IN "system";
COMMENT ON COLUMN "DBA"."articulos"."impuesto_interno_porcentaje" IS '%
impuesto interno';
COMMENT ON COLUMN "DBA"."articulos"."id_iva" IS '0 exento; 1- (21%);
2-(10.5%)';

CREATE TABLE "DBA"."secciones" (
    "codigo" NUMERIC(5,0) NOT NULL,
    "descripcion" CHAR(35) NULL,
    "rubro" NUMERIC(2,0) NOT NULL,
    "seccion_contab" NUMERIC(3,0) NULL,
    "habil" CHAR(1) NULL,
    "sucursal" INTEGER NULL,
    "es_viajante" CHAR(1) NOT NULL DEFAULT '0',
    PRIMARY KEY ( "codigo" ASC, "rubro" ASC )
) IN "system";

CREATE TABLE "DBA"."departamentos" (
    "codigo" NUMERIC(5,0) NOT NULL,
    "descripcion" CHAR(35) NULL,
    "seccion" NUMERIC(3,0) NOT NULL,
    "rubro" NUMERIC(2,0) NOT NULL,
    "habil" CHAR(1) NOT NULL,
    PRIMARY KEY ( "codigo" ASC, "seccion" ASC, "rubro" ASC )
) IN "system";

CREATE TABLE "DBA"."grupos" (
    "codigo" NUMERIC(5,0) NOT NULL,

```

```

        "descripcion" CHAR(35) NULL,
        "departamento" NUMERIC(4,0) NOT NULL,
        "seccion" NUMERIC(3,0) NOT NULL,
        "rubro" NUMERIC(2,0) NOT NULL,
        "habil" CHAR(1) NOT NULL DEFAULT 'N',
        PRIMARY KEY ( "codigo" ASC, "departamento" ASC, "seccion" ASC,
"rubro" ASC )
    ) IN "system";

CREATE TABLE "DBA"."marcas" (
    "codigo" DECIMAL(6,0) NOT NULL,
    "descripcion" VARCHAR(30) NULL,
    "habil" CHAR(1) NULL,
    "rubro" INTEGER NOT NULL,
    "seccion" INTEGER NULL,
    PRIMARY KEY ( "codigo" ASC, "rubro" ASC )
) IN "system";

```