******

**Análisis de las ventas compras y stock**

Unamuno Acha, Isabel

2023 - 2024

Máster Universitario en

Análisis de Datos para la Inteligencia de Negocio / Business Analytics

TRABAJO DE FIN DE MÁSTER

**ANÁLISIS DE INVENTARIO**

TRABAJO DE FIN DE MÁSTER PRESENTADO EN: Mondragon Unibertsitatea

PARA LA OBTENCIÓN DEL TÍTULO DE: Máster Universitario en Análisis de Datos para la Inteligencia de Negocio / Business Analytics

AUTOR/A: Isabel Unamuno Acha

DIRECTOR/A: XXX

TUTOR/A: Jon Perez Visaires

ORGANIZACIÓN EN LA QUE HA REALIZADO EL PROYECTO: Datua

FECHA DE DEFENSA: Bilbao, a 05 de 07 de 2024

El autor/la autora del Trabajo de Fin de Máster, autoriza a la Facultad de Empresariales de Mondragon Unibertsitatea, con carácter gratuito y con fines exclusivamente de investigación y docencia, los derechos de reproducción y comunicación pública de este documento siempre que: se cite el autor/la autora original, y el uso que se haga de la obra no sea comercial.

****

**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

DECLARACIÓN DE ORIGINALIDAD

Yo Isabel Unamuno Acha

Declaro que este Trabajo de Fin de Máster es original, fruto de mi trabajo personal, y que no ha sido previamente presentado para obtener otro título o calificación profesional.

Las ideas, formulaciones, imágenes, ilustraciones tomadas de fuentes ajenas han sido debidamente citadas y referenciadas.

**RESUMEN**

**LABURPENA**

**ABSTRACT**

**ÍNDICE**

RESUMEN 3

LABURPENA 4

ABSTRACT 5

1. INTRODUCCIÓN 10

1.1. Presentación de la empresa y problemática 10

1.2. Marco teórico 10

1.2.1. Red Neuronal Artificial multicapa con Embeddings 12

1.2.2. Gradient Boosting Regressor 12

1.2.3. Random Forest Regressor 12

1.2.4. Decision Tree Regressor 12

1.2.5. Light Gradient Boosting Machine Regressor 12

1.3. Objetivos 15

1.4. Descripción de la metodología empleada 16

1.5. Planificación 17

2. DESARROLLO 18

2.1. Análisis de los datos 18

2.1.1. Entrada de datos: 18

2.1.2. Análisis exploratorio 19

2.1.3. Limpieza y procesamiento 19

2.1.3.1. Tratamiento de *Missings* 19

2.1.3.2. Tratamiento de duplicados 20

2.1.3.3. Irregularidades en los datos 21

2.2. Modelado de datos 22

2.2.1. Modelo semántico 22

2.2.2. Relaciones entre tablas de hechos y dimensiones 23

2.2.3. Definición de medidas y KPIs 27

2.2.3.1. Medidas 27

2.2.3.2. Tablas calculadas 27

2.3. Visualización 28

2.3.1. Estructura del informe 28

2.3.2. Especificación del diseño del informe analítico 28

2.3.2.1. Espacio, márgenes y equilibrio 28

2.3.2.2. Tamaño del objeto visual 29

2.3.2.3. Alineación y tamaño del objeto visual 29

2.3.2.4. Color 29

2.3.2.5. Coherencia en el diseño de informes 29

2.3.2.6. Selección de objetos visuales en informes 29

2.3.2.7. Formato y configuración de visualizaciones 30

2.3.2.8. Trabajo con marcadores 31

2.4. Aplicación de Modelos para la predicción de la demanda 32

2.1. Control de versiones en GIT 37

3. RESULTADOS 39

4. CONCLUSIONES Y DISCUSIÓN 40

**ÍNDICE DE FIGURAS**

Figura 2: Modelo estrella 23

Figura 1:Descripción del uso de Power BI para crear análisis controlados por datos 26

Figura 3: Diseño del modelo 26

Figura 4:Serie temporal de ventas y compras 32

Figura 5: Ventas diarias 35

Figura 6: Ventas diarias con y sin estacionalidad 36

*Figura 7: Predicción de ventas sin estacionalidad* 36

**ÍNDICE DE ANEXOS**

[Anexo A: Descripción de las tablas del modelo 43](#_Toc169262034)

[Anexo B: Descripción de las medidas calculadas 45](#_Toc169262035)

# INTRODUCCIÓN

## Presentación de la empresa y problemática

El entorno empresarial actual se caracteriza por un creciente control basado en datos. Tanto pequeñas como grandes empresas utilizan los datos para decisiones en ventas, contratación, objetivos y otras áreas. Sin embargo, para el usuario promedio sin conocimientos de análisis de datos o estadísticas, puede ser difícil comprender y visualizar estos datos de manera clara.

Eso es lo que le ocurre a la siguiente empresa. Es un mayorista de vinos y licores que opera en múltiples ubicaciones en Estados Unidos. Con aproximadamente 80 establecimientos y unas ventas anuales superiores a 450 millones de dólares, la empresa ha experimentado dificultades significativas para gestionar el alto volumen de datos sobre las compras, ventas y stock. La empresa maneja ventas y costos de bienes vendidos que alcanzan los cientos de millones de dólares, lo que hace que las hojas de cálculo tradicionales resulten insuficientes para gestionar el vasto volumen de datos, que incluye millones de registros de ventas, compras e inventario. Esta complejidad requiere un enfoque sofisticado para un análisis eficaz. Estos problemas han llevado al equipo directivo a buscar soluciones para analizar y optimizar sus prácticas de gestión. El objetivo es diseñar un cuadro de mando que permita controlar las compras, ventas y el stock mediante una interacción rápida y comprensible, así como obtener información valiosa sobre clientes y proveedores. Esto permitirá a la empresa tomar decisiones de valor de un vistazo, mejorando la eficiencia operativa y la capacidad de respuesta en un entorno altamente competitivo.

## Marco teórico

En el entorno empresarial actual, la aplicación de algoritmos de *Machine Learning (ML)* para la gestión de inventarios se ha convertido en una herramienta esencial. Estas técnicas avanzadas permiten a las empresas predecir con mayor precisión la demanda futura y ajustar sus niveles de stock de manera eficiente. Al optimizar los costos de almacenamiento y reducir las pérdidas asociadas con la escasez de productos, las empresas pueden no solo mejorar su rentabilidad, sino también elevar la satisfacción del cliente al asegurar la disponibilidad continua de los productos más demandados. La implementación de algoritmos de *ML* en la gestión de inventarios proporciona una ventaja competitiva significativa. Estas herramientas permiten a las empresas operar de forma más eficiente, responder mejor a las fluctuaciones del mercado y adaptarse rápidamente a las necesidades cambiantes de los consumidores. Por ello, en el siguiente apartado, primero de todo se ha resumido el concepto de *ML* y se han analizado cuáles son los diferentes algoritmos que se emplean actualmente para la predicción de la demanda y la mejora en la gestión de *stock*.[1]

Los algoritmos de *ML* permiten identificar patrones complejos en grandes volúmenes de datos, infiriendo sus propias reglas para reconocer patrones similares en nuevos conjuntos de datos. Esto da lugar a la creación de sistemas inteligentes que mejoran de forma autónoma mediante la observación de datos. Estos sistemas pueden aprender a predecir comportamientos, detectar similitudes o anomalías de manera automática, y tomar decisiones adecuadas basadas en los datos analizados. Los modelos de *ML* analizan datos reales que representan el proceso que se desea mejorar o automatizar. Este proceso, conocido como entrenamiento de algoritmos, permite que el sistema extraiga conclusiones relevantes y aprenda a realizar tareas sin necesidad de programación previa, simplemente observando y analizando los datos proporcionados. En los proyectos de *ML*, hay tres pasos que se deben seguir, que son los siguientes.

* Revisión de datos: Implica examinar los datos disponibles para determinar cuáles son útiles, asegurando su correcto almacenamiento y formatos adecuados. Es crucial limpiar los datos eliminando registros antiguos, incompletos o erróneos para garantizar la calidad del *dataset*.
* Organización de datos: Fundamental para facilitar el aprendizaje automático. Se seleccionan datos que reflejen las decisiones que el sistema deberá tomar automáticamente, estructurándolos de manera que optimicen el proceso de entrenamiento del modelo.
* Entrenamiento y la validación del modelo: Durante esta fase, el modelo analiza los datos para identificar patrones y relaciones, ajustando sus parámetros para poder detectar estas características automáticamente en futuros datos. Este proceso permite al modelo configurar su propio sistema de detección basado en la información observada.[2]

Tras conocer los conceptos generales del *ML*, se ha analizado como se aplican en la gestión de inventarios actualmente. Las redes neuronales son ampliamente utilizadas para predecir el comportamiento del inventario. Los modelos de *backpropagation* (BPNN) destacan por su estructura simple y capacidad de aprendizaje. Entre los científicos que han desarrollado varios artículos, se han mencionado algunos que se han considerado interesantes. Sustrova desarrolló varios modelos de redes neuronales para optimizar la cantidad de stock, seleccionando el modelo con menor error cuadrático medio (MSE) y un coeficiente de determinación (R2) cercano a 1. Su modelo final fue una red feed-forward con backpropagation, función de entrenamiento TRAINGDX y función de transferencia TANSIG [3]. Shoujing Zhang combinó BPNN con clustering mejorado y PCA para gestionar el inventario de piezas de repuesto en mantenimiento de camiones, evitando el impacto subjetivo de los gestores de inventario[4]. Por otro lado, Praveen K.B. y otros utilizaron el modelo de regresión XGBoost para predicciones de demanda en pequeñas y medianas empresas, logrando reducir el stock y el capital dedicado[5]. Se han recogido algunos de los algoritmos que se han considerado interesantes y se ha explicado la función de ellas brevemente.

### Red Neuronal Artificial multicapa con Embeddings

Los *embeddings* son una técnica para convertir variables categóricas en representaciones continuas, que se utilizan como entradas en un modelo de red neuronal para aprendizaje supervisado. Estas representaciones permiten que las categorías similares se ubiquen más cerca entre sí y se ajustan durante el entrenamiento para minimizar la pérdida del modelo. Al emplear *embeddings*, se pueden obtener predicciones más precisas en comparación con el uso exclusivo de las unidades de stock[1].

### Gradient Boosting Regressor

Este estimador construye un modelo aditivo por etapas y permite optimizar funciones de pérdida diferenciables arbitrarias. En cada etapa se ajusta un árbol de regresión sobre el gradiente negativo de la función de pérdida dada[6].

### Random Forest Regressor

Un bosque aleatorio es un meta estimador que ajusta una serie de árboles de decisión regresores a varias submuestras del conjunto de datos y utiliza el promedio para mejorar la precisión predictiva y controlar el sobreajuste.

### Decision Tree Regressor

Un árbol de decisión es un tipo de algoritmo de aprendizaje supervisado que se usa comúnmente en el aprendizaje automático para modelar y predecir resultados basados en datos de entrada. Es una estructura en forma de árbol donde cada nodo interno prueba el atributo, cada rama corresponde al valor del atributo y cada nodo hoja representa la decisión o predicción final. El algoritmo del árbol de decisión entra en la categoría de aprendizaje supervisado. Se pueden utilizar para resolver problemas tanto de regresión como de clasificación[7].

### Light Gradient Boosting Machine Regressor

Es un marco de código abierto ampliamente utilizado para el refuerzo por gradiente. Destacado por su capacidad para manejar conjuntos de datos de gran tamaño, ofrece un rendimiento superior en comparación con otros marcos de refuerzo por gradiente como XGBoost y CatBoost. Una de las características distintivas de Light GBM es su método de muestreo unilateral basado en el gradiente para dividir los árboles. Este enfoque reduce significativamente el uso de memoria y mejora la precisión del modelo. Además, Light GBM emplea un crecimiento por hojas en lugar de crecimiento por niveles, lo que lo hace considerablemente más rápido que los métodos tradicionales de crecimiento en profundidad[8].

Volviendo al entorno empresarial, vivimos en un momento de rápidos cambios en el área tecnológica, donde todas las empresas deben poder reaccionar rápidamente. Cualquier organización necesita adaptarse cada vez más rápido a un mundo en constante cambio. Además, cada día aumenta la complejidad de la cadena de suministro. Por lo tanto, no hay otra manera de que las empresas sobrevivan y se adapten rápidamente a esos cambios que automatizando procesos en su gestión de suministros. Si bien no se pueden resolver todos los desafíos simplemente añadiendo tecnología, una estrategia de cadena de suministro moderna requiere una pila de tecnología actualizada. Las cadenas de suministro, tradicionalmente lineales y predecibles, enfrentan una importante fragmentación de la demanda debido al crecimiento del comercio digital y nuevos modelos de cumplimiento. Esta evolución está haciendo que las cadenas actuales se vuelvan obsoletas. Aunque nadie puede prever con certeza el futuro de la cadena de suministro, es claro que la inteligencia artificial jugará un papel fundamental para impulsar su resiliencia. Se espera una mayor visibilidad de extremo a extremo con sistemas más interconectados y un mayor uso de la IA y el *ML* para la previsión de la demanda y prácticas más sostenibles, respondiendo a las demandas de los consumidores. En un futuro cercano, las cadenas de suministro podrían ser altamente autónomas, con sistemas impulsados por IA que gestionen la mayoría de los procesos, desde la adquisición hasta la entrega[9].

El *ML* ha demostrado tener un gran potencial en varios ámbitos, incluida la configuración de optimización del control de inventario. La optimización de inventario es fundamental para la gestión eficiente de los minoristas, independientemente de su tamaño. Implica manejar una gran cantidad de productos de manera regular con el objetivo de reducir costos operativos y aumentar las ventas. Una parte esencial de este proceso es el control de inventario, que implica decidir cuándo y cuánto pedir de un artículo en particular para mantener un equilibrio óptimo entre oferta y demanda. Para ello, es esencial optimizar las políticas de inventario, ajustándolas según parámetros como el periodo de revisión, el tiempo de entrega y el nivel de servicio objetivo para cada artículo. El *ML*, al aprovechar algoritmos avanzados para analizar datos, Puede manejar patrones de demanda complejos, identificar tendencias y ajustar pronósticos según múltiples variables y restricciones. Esto mejora la precisión de la previsión de la demanda y optimiza los niveles de inventario en consecuencia. Con su capacidad para aprender y adaptarse a partir de datos, el *ML* es más dinámico y flexible que el software tradicional de gestión de inventario, especialmente en la previsión y optimización de la demanda. Esto permite a las empresas determinar puntos de reabastecimiento óptimos, niveles de existencias de seguridad y estrategias de asignación de inventario, lo que resulta en decisiones de gestión de inventario más precisas, una mejor gestión de la cadena de suministro, y una reducción del riesgo de desabastecimiento o exceso de inventario.

Implementar *ML* en la gestión de inventario ofrece una serie de beneficios significativos: Por un lado, la mejora de la precisión de pronósticos. Los algoritmos permiten determinar niveles óptimos de inventario considerando factores como tiempo de entrega, estacionalidad y limitaciones de costos. Identifican el equilibrio adecuado entre costos de mantenimiento y desabastecimientos, optimizando los niveles de inventario a través del análisis de datos históricos, ciclos de producción y pronósticos de ventas. Por otro lado, la reducción de costos y pérdidas, es decir, analizan datos históricos de ventas, tendencias del mercado y factores externos para pronosticar con precisión la demanda de los clientes. Permite generar pronósticos de demanda más precisos, optimizando los niveles de inventario, reduciendo desabastecimientos y evitando excesos de inventario. Por último, también optimiza el ciclo de vida del producto. Los algoritmos de *ML* pueden generar probabilidades asociadas con diferentes niveles de demanda, permitiendo una gestión más eficiente de los productos. Por ejemplo, en una tienda de comestibles, pueden analizar fechas de vencimiento, patrones de demanda y datos históricos de ventas para optimizar los niveles de stock y minimizar el desperdicio debido a la caducidad del producto.

Estas son varias consideraciones y pasos clave para implementar *ML* en la gestión de inventario.

* La evaluación de necesidades, es decir, reflexionar sobre objetivos claros y específicos, como evitar el exceso de existencias, prevenir desabastecimientos o mejorar la previsión de la demanda.
* La selección de modelos y algoritmos, seleccionado modelos como regresión lineal, árboles de decisión o redes neuronales según la complejidad del problema y los datos disponibles.
* Entrenar los modelos con datos históricos, evaluando su rendimiento con métricas adaptadas a los objetivos de gestión de inventario, y ajustar y optimizar el modelo para alcanzar un rendimiento óptimo.
* La integración con sistemas existentes, colaborando estrechamente entre analistas de datos, profesionales de IT y expertos en gestión de suministros para integrar el modelo en el sistema de gestión de inventario.
* Alinear los resultados del modelo con los procesos existentes y garantizar una integración fluida en la toma de decisiones.
* Establecer un proceso de monitoreo y mantenimiento constante, dado que los modelos de *ML* son dinámicos y requieren actualizaciones periódicas.

Empresas líderes como Amazon, Walmart y Nike han implementado con éxito el *ML* en la gestión de inventario, obteniendo resultados significativos. En el caso de Amazon, por ejemplo, predice la demanda de productos considerando diversas variables como tendencias de búsqueda, datos históricos de ventas y condiciones climáticas. Esto le permite mantener niveles óptimos de inventario, evitando excesos o faltantes de stock. Para el caso de Walmart, emplea un sistema de gestión de inventario impulsado por IA para proporcionar a los clientes lo que necesitan, cuando lo necesitan y al costo esperado. Combina datos históricos con análisis predictivos para colocar estratégicamente los artículos en centros de distribución y tiendas, optimizando la experiencia de compra. Y por último, Nike, está construyendo una cadena de suministro digital a nivel mundial para atender directamente a los consumidores a escala[10].

## Objetivos

El objetivo de este proyecto es desarrollar un cuadro de mando integral que permita visualizar la información clave en tiempo real para mejorar las prácticas de gestión de inventarios, control de compras y ventas, y seguimiento de clientes potenciales en la empresa. Este *dashboard* centralizará los datos esenciales, facilitando la toma de decisiones informadas y estratégicas para optimizar las operaciones y aumentar la eficiencia de la gestión empresarial. Con esta herramienta, se busca asegurar una mejor planificación, reducir costos, mejorar el servicio al cliente y maximizar los ingresos. Las tareas del proyecto se dividen en dos objetivos principales:

* Realizar un preprocesamiento efectivo de los datos de inventario utilizando **Python** para garantizar la calidad y la integridad de los datos antes de su análisis.
* Implementar scripts de Python que limpien, formateen y preparen los datos de inventario para su análisis.
* Utilizar bibliotecas de Python como Pandas y NumPy para realizar el preprocesamiento de datos de manera eficiente.
* Garantizar que los datos estén libres de errores y sean coherentes para un análisis preciso.
* Crear un informe interactivo en Power BI que aborde los problemas de gestión de inventario identificados y proporcione una visión clara del estado del inventario en todo momento.
* Desarrollar un informe en Power BI que incluya visualizaciones dinámicas de datos de inventario, métricas clave y recomendaciones para mejorar la gestión de inventario.
* Utilizar las capacidades de Power BI para crear visualizaciones efectivas y personalizadas que aborden los problemas específicos de gestión de inventario.
* Implementar un control de versiones utilizando GitHub para gestionar y rastrear cambios en el código y los recursos del proyecto de análisis de inventario.
* Crear un repositorio en GitHub y realizar commits periódicos para registrar los cambios en el código y los archivos relacionados con el proyecto.

## Descripción de la metodología empleada

En términos de herramientas de software, este proyecto sigue una metodología centrada en el uso de Python para el procesamiento de datos. Se emplea un entorno Jupyter Notebook (.ipynb) para llevar a cabo todas las fases del tratamiento de datos, incluyendo carga, preprocesamiento y modelado. Además, para la creación del cuadro de mando, se utiliza Power BI. Estas herramientas han sido seleccionadas debido a su idoneidad para un proyecto nuevo que se desarrolla completamente desde cero. En relación con las diferentes etapas que abarca este proyecto, se pueden distinguir las siguientes:

* Carga de datos: En la fase inicial del proyecto, los datos se recopilan inicialmente en archivos en formato CSV o XLSX. Estos archivos se transforman en *Dataframes* de Pandas utilizando *Python* dentro del entorno *Jupyter Notebook*.
* Preprocesamiento de los datos: Durante esta etapa inicial, se lleva a cabo el procesamiento de los datos con el objetivo de asegurar su calidad y prepararlos adecuadamente para las tareas de análisis. Esto incluye la limpieza de los datos, el tratamiento de valores nulos y la normalización de datos cuando es necesario
* Análisis de los datos: En la fase de análisis de datos, se ha realizado un esfuerzo por comprender exhaustivamente los datos disponibles. Se han generado nuevas tablas que añaden valor al informe y se han creado columnas adicionales a partir de los datos existentes para enriquecer la información disponible
* Creación del informe: Las bases de datos almacenadas en Excel han sido integradas en Power BI, donde se ha procedido a crear un informe dinámico y claro. Este proceso asegura la correcta visualización de los datos de manera accesible y comprensible para los usuarios.
* Control de versiones: Durante todo el proyecto se ha hecho uso se git para mantener una correcta gestión de las versiones.
* Conclusiones: una vez completado todo el proceso se extraen las conclusiones definitivas.

## Planificación

(Insertar Gantt)

# DESARROLLO

Para llevar a cabo el proyecto se ha cogido la información de una plataforma llamada Kaggle que proporciona un entorno colaborativo para el aprendizaje automático y la ciencia de datos. Esta página ofrece una variedad de recursos y herramientas en los que entre otros se encuentran bases de datos públicos[11].

En el desarrollo del proyecto se han dado una serie de pasos interrelacionados que van desde la entrada de datos hasta la visualización de los resultados. En el primer paso se han ingestado los datos, lo que implica la recopilación de información relevante. A continuación, se ha seguido con la preparación de datos, incluyendo la limpieza y consolidación de datos brutos para transformarlos en una forma adecuada para el análisis. Verificando continuamente para garantizar la calidad y relevancia de los datos. Luego, en la exploración de datos se ha enriquecido la base de datos con información adicional para proporcionar una perspectiva más completa. Por último, se ha generado el informe donde se organizan y presentan los resultados del análisis de manera comprensible y útil para compartir los conocimientos adquiridos.

## Análisis de los datos

En esta fase se ha realizado lo que sería el proceso ETL, es decir, extracción de datos desde origen, carga de datos, transformación de datos y la nomenclatura de tablas y columnas.

### Entrada de datos:

A la hora de leer los datos y el procesarlos, se ha optado por emplear el entorno de desarrollo integrado *Visual Studio Code*, aprovechando su funcionalidad y facilidad de uso en la programación en Python, un lenguaje ampliamente utilizado en el ámbito del análisis y ciencia de datos. En la fase inicial del proceso, se ha procedido a importar las bibliotecas y paquetes necesarios para la manipulación, análisis y visualización de datos. Entre estas herramientas esenciales, se incluyen las siguientes:

* Pandas, una biblioteca de manipulación y análisis de datos que proporciona estructuras de datos flexibles y eficientes, así como funciones para la lectura y escritura de datos en diversos formatos[12].
* NumPy, una biblioteca fundamental para la computación numérica en Python, que ofrece soporte para matrices y funciones matemáticas de alto nivel[13].
* *Matplotlib*, una biblioteca ampliamente utilizada para la visualización de datos en dos dimensiones, que permite la creación de gráficos estáticos, interactivos y animados[14].
* Seaborn, una biblioteca basada en *Matplotlib* que proporciona una interfaz de alto nivel para la creación de gráficos estadísticos atractivos e informativos[15].
* *PyExcel* una biblioteca de Python diseñada para facilitar la lectura, escritura y manipulación de hojas de cálculo en diferentes formatos, como Excel (.xlsx, .xls).

La elección de estas herramientas se sustenta en su capacidad para facilitar las tareas de lectura, procesamiento y análisis de datos, así como en su versatilidad y potencia para generar visualizaciones claras y efectivas que permitan comprender y comunicar adecuadamente los resultados obtenidos durante el análisis de datos. Dicho esto, se han cargado los seis archivos XLSX con los que se dispone para llevar adelante el trabajo. Estos archivos corresponden a operaciones comerciales del ejercicio cerrado en 2016. El conjunto de datos incluye las siguientes bases de datos: Inventario inicial del 2016, inventario final del 2016, facturas de compra correspondientes al 2016, precios de compra, datos de compras y datos de ventas. Una vez cargados los datos se han impreso las primeras 5 líneas de todos los datos para obtener una visión general de los datos.

### Análisis exploratorio

Se ha realizado una función que devuelve una tabla, como la que se encuentra en la (Tabla 1) donde se han recogido detalles clave sobre los archivos leídos. Entre ellos, los nombres de las columnas, el número de filas, el tipo de los datos, el número de valores únicos de cada columna, los valores no informados y el porcentaje de valores no informados frente al total. encuentra Así con la foto clave de cada tabla se han sacado varias conclusiones para después proceder a la limpieza y preprocesamiento de los datos.

Tabla 1: Análisis exploratorio de los datos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Columna** | **TipoDato** | **Filas** | **Valores Unicos** | **Missings** | **Missings (%)** |
| 0 | InventoryId | object | 224489 | 224489 | 0 | 0.000 |
| 1 | Store | int64 | 224489 | 80 | 0 | 0.000 |
| 2 | City | object | 224489 | 67 | 1284 | 0.572 |
| 3 | Brand | int64 | 224489 | 9653 | 0 | 0.000 |
| 4 | Description | object | 224489 | 8732 | 0 | 0.000 |
| 5 | Size | object | 224489 | 47 | 0 | 0.000 |
| 6 | onHand | int64 | 224489 | 548 | 0 | 0.000 |
| 7 | Price | float64 | 224489 | 354 | 0 | 0.000 |
| 8 | endDate | object | 224489 | 1 | 0 | 0.000 |

Por un lado, se ha observado que la columna *InventoryId* tiene los mismos registros únicos que los totales, lo que sugiere que podrían servir como claves primaria y foránea para unir las tablas. Por otro lados, que las tablas contienen valores no informados que son necesarios identificar y tratar adecuadamente. Además, las columnas de tamaño muestran incoherencias en todas las tablas debido a la variación de unidades de medida. Por ello, es necesario convertirlas a una unidad única y coherente. Aparte de la columna de volumen, en alguna de las tablas también existe la columna *Size*, la cual contiene el mismo dato, por lo que se puede eliminar una de las dos columnas. En cuanto al formato de las fechas se ha observado que utilizan diferentes formatos en cada una de las tablas por lo que es conveniente unificar y ponerlos de la misma manera. Por último, en la columna *VendorName* se pueden ver espacios o signos tras la palabra que se pueden eliminar.

### Limpieza y procesamiento

En el apartado anterior se han mencionado cuales son los puntos que se deben abordar para llevar a cabo la limpieza y el procesamiento de los datos, por lo que a continuación se han producido dichos cambios.

#### Tratamiento de *Missings*

La integridad y la calidad de los datos son aspectos fundamentales en cualquier análisis de datos. Los valores no informados, comúnmente conocidos como *missing* *values* en inglés, representan una preocupación significativa en el contexto del análisis de datos, ya que pueden distorsionar los resultados y conducir a interpretaciones erróneas si no se manejan adecuadamente. Por lo tanto, es crucial detectar y abordar de manera efectiva estos valores faltantes. Los valores no informados pueden surgir debido a una variedad de razones, que van desde errores en la entrada de datos hasta fallos en la recopilación o transferencia de información. Independientemente de su origen, la presencia de valores no informados puede comprometer la validez y la fiabilidad de cualquier análisis realizado sobre los datos afectados. Durante el análisis del conjunto de, se han detectado valores no informados en cuatro de las tablas analizadas. A continuación, se describen las acciones tomadas para abordar estos valores faltantes:

* Tabla inventario final: En la columna *City* se ha identificado ausencia de valores. Tras un análisis detallado, se ha observado que todos los nombres de ciudades están presentes excepto para una ubicación de tienda específica. Es decir, todos los nombres faltantes en la columna ciudad pertenecen al número de tienda 46. Siendo esto así, se ha observado en la tabla inventario inicial donde se encuentra dicha tienda obteniendo el nombre *TYWARDREATH* como respuesta. Por lo que se ha decidido reemplazar los valores nulos por esa ciudad.
* Tabla compras: Se han encontrado solo tres valores faltantes en la columna *Size*, lo que representa un porcentaje mínimo del total de registros. Por lo tanto, se ha optado por eliminar estas filas para preservar la integridad de los datos restantes.
* Tabla ventas: En la columna *Approval* se muestra un alto porcentaje de valores faltantes, lo que representa el 93% del total de registros. Dada la magnitud de esta ausencia de información, se ha tomado la decisión de eliminar completamente esta columna para evitar cualquier sesgo o distorsión en el análisis posterior.
* Tabla precio compras: Se han identificado valores faltantes en las columnas *Description*, *Size* y *Volume*. Dado que estos valores no informados representan registros individuales, se ha decidido eliminar las filas correspondientes para mantener la coherencia y la integridad de los datos restantes.

Una vez tratados los *missings* de cada tabla se ha creado una función que devuelve la cantidad de valores no informados para cada tabla y se ha obtenido un valor de 0 para cada una de ellas.

#### Tratamiento de duplicados

Mediante un bucle y el método *duplicated* en Python, se ha realizado una verificación para determinar la presencia de duplicados dentro del conjunto de datos representado. Se ha guardado la información en una variable llamada duplicados, la cual captura el resultado booleano, proporcionando una indicación clara sobre la presencia o ausencia de duplicados en los datos. En este caso ninguna de las tablas contaba con valores duplicados, por lo que no se ha aplicado ninguna modificación.

#### Irregularidades en los datos

Se han observado irregularidades significativas en las entradas de datos, especialmente en la columna *Size*, que exhibe variaciones en todas las tablas de datos. Estas discrepancias se refieren a la representación de información de volumen, expresada en diferentes unidades y formatos, como litros, mililitros, onzas, paquetes (pk) y combinaciones de estas unidades. Este panorama heterogéneo implica la necesidad de estandarizar estas medidas en una unidad de volumen única y coherente. Para abordar esta cuestión, se ha llevado a cabo un análisis exhaustivo de los recuentos únicos en la columna *Size*. Por un lado, aquellos elementos que exhiben un solo recuento y no presentan un patrón discernible han sido asignados a una tasa equivalente estándar. Por otro lado, aquellos elementos con múltiples recuentos y un patrón discernible han sido transformados de acuerdo con estos patrones identificados.

Este enfoque metodológico busca garantizar la consistencia y la comparabilidad de los datos de volumen a lo largo de las diferentes tablas, lo que es esencial para facilitar un análisis coherente. La estandarización de las unidades de volumen contribuye a mitigar posibles sesgos o distorsiones en los resultados del análisis, permitiendo así una interpretación precisa y fiable de los datos. También se ha creado una función para eliminar los espacios finales en las columnas especificadas. Además, para garantizar la coherencia y la fiabilidad en el manejo de datos temporales, se ha creado una función que se encarga de estandarizar las fechas en varios *DataFrames*, convirtiéndolas al formato día/mes/año para facilitar su lectura y manejo posterior. Al estandarizar el formato de fecha y hora en todas las columnas pertinentes del *DataFrame*, se promueve la precisión y la interpretabilidad de los datos, lo que es esencial para obtener conclusiones sólidas y confiables en cualquier análisis posterior.

Por último, se ha identificado una discrepancia en los nombres de las ciudades presentes en las tablas de inventario inicial e inventario final, donde los nombres no corresponden a ubicaciones geográficas reales. Para abordar esta irregularidad, se ha realizado un proceso de corrección mediante el reemplazo de los nombres de las ciudades no reales por aquellos que sí lo son. Para llevar a cabo esta corrección, se han extraído todos los nombres de ciudades presentes en las tablas y se han reemplazado con una lista de ciudades geográficamente válidas. Para complementar la información, se ha añadido una nueva columna llamada estado que indica el estado de cada ciudad. Esta información adicional se ha considerado apropiada para visualizarlo en el informe mediante un mapa geográfico. Además, se ha observado que el identificador de inventario (*InventoryId*) estaba influenciado por el nombre de la ciudad. Por lo tanto, se ha procedido a recrear este identificador combinando el número de tienda, el nombre de la ciudad y el número de marca, separados por un guion. Así, garantiza que los identificadores de inventario reflejen de manera precisa la relación entre la tienda, la ciudad y la marca, proporcionando así una representación más fiel de la realidad en el conjunto de datos. Para terminar, se han impreso otra vez todas las tablas, y se ha asegurado que se ha completado la limpieza y procesamiento de los datos, para así empezar con la siguiente fase.

## Modelado de datos

Una vez limpiados y procesados los datos, se ha comenzado con el modelado de datos. Esta es una parte fundamental en un proyecto BI cuyo propósito es organizar y estructurar los datos de manera que permite al cliente realizar el análisis necesario. En este proyecto se quieren mostrar las ventas, compras y stock de la empresa, por lo que para realizar el modelado de datos es importante identificar todas las métricas a analizar, para posteriormente proceder a definir el modelo de datos de la forma óptima.

### Modelo semántico

Este método organiza los datos de manera lógica, específica y fácilmente comprensible. Puede entenderse como una forma de dar orden y contexto a los datos. El modelo semántico organiza los elementos de datos describiendo las relaciones entre ellos. Un buen modelo semántico ofrece las siguientes ventajas: una exploración de datos más rápida, agregaciones más fáciles de crear, informes más precisos, dedicar menos tiempo a crear el informe y mejor seguimiento a futuro[16].

A la hora de crear el modelo, se ha optado por el modelo estrella, en el que los datos se organizan en hechos (elementos medibles) y dimensiones (información de referencia), donde cada hecho está rodeado por sus dimensiones asociadas en un patrón similar a una estrella como se puede ver en la Figura 2. También existe el esquema de copo de nieve, que se parece al esquema de estrella, pero incluye capas adicionales de dimensiones asociadas, lo que hace que la ramificación del patrón sea más compleja.

Figura 2: Modelo estrella

Diagrama, Forma

Descripción generada automáticamente con confianza media

Fuente: [16]

El esquema en estrella se fundamenta en una tabla de hechos que contiene uno o más datos que en lo posible deben ser medibles y una o más tablas de dimensiones. La llave primaria de la tabla de hechos es una concatenación de las llaves primarias de cada una de las dimensiones. Los hechos, con frecuencia datos agregados, pueden identificarse como medidas tomadas de la intersección de todas las dimensiones. Los atributos en las dimensiones son, por lo general, textuales y discretos, con el propósito de establecer restricciones específicas en las consultas. La construcción de una base de datos bajo este esquema requiere de una rígida definición de hechos y dimensiones que anticipen la consulta de datos bajo patrones preestablecidos [17].

Las tablas de hechos contienen valores de datos de eventos o de observación y pueden contener varios valores repetidos. Las tablas de dimensiones contienen los detalles sobre los datos de las tablas de hechos: productos, ubicaciones, clientes y proveedores. Estas tablas están conectadas a la tabla de hechos a través de columnas de clave. Las tablas de dimensiones se usan para filtrar y agrupar los datos de las tablas de hechos [18].

### Relaciones entre tablas de hechos y dimensiones

Al tener varias tablas, las relaciones entre ellas son necesarias para calcular los resultados de forma precisa y mostrar la información correcta en los informes. En caso de que se hayan nombrado correctamente las columnas, es decir, poniendo el mismo nombre a aquellas que se quieren relacionar, la característica de detección automática es capaz de crear las relaciones automáticamente. Sin embargo, es necesario revisarlas todas y realizar cambios en los casos que sea necesario. Además, al crear o editar una relación, se puede configurar más de una opción. La opción cardinalidad puede tener una de las siguientes configuraciones:

* Varios a uno (\*:1): Indica que una columna de una tabla puede contener múltiples instancias de un valor, mientras que la tabla relacionada, frecuentemente denominada tabla de búsqueda, tiene solo una instancia de ese valor.
* Uno a uno (1:1): La columna de una tabla contiene una única instancia de un valor específico, y la tabla relacionada también contiene una única instancia de ese mismo valor.
* Uno a varios (1: \*): En una relación de uno a varios, la columna de una tabla contiene una sola instancia de un valor, y la tabla relacionada puede contener múltiples instancias de ese mismo valor.
* Varios a varios (:): Los modelos compuestos permiten establecer relaciones de varios a varios entre tablas, eliminando la necesidad de valores únicos en las tablas. Esto también facilita evitar soluciones alternativas previas, como la creación de nuevas tablas únicamente para establecer relaciones.

La opción dirección de filtro cruzado también es muy importante a la hora de crear las relaciones que puede tener unas de las siguientes opciones

* Ambos: Esta configuración indica que, para efectos de filtrado, ambas tablas se tratan como si fueran una sola tabla. Se utiliza cuando una única tabla rodeada de muchas tablas de búsqueda.
* Único: Esta es la dirección predeterminada más común, donde las opciones de filtrado en las tablas conectadas afectan solo a la tabla en la que se agregan los valores[18].

Por último, recalcar la importancia de las claves primarias y foráneas en el contexto de Power BI para definir y gestionar las relaciones entre las tablas de un modelo de datos. Una clave primaria es una columna o un conjunto de columnas en una tabla cuyos valores identifican de forma exclusiva una fila de la tabla. Una base de datos relacional está diseñada para imponer la exclusividad de las claves primarias permitiendo que haya sólo una fila con un valor de clave primaria específico en una tabla. En este caso las claves primarias se han nombrado indicando un id por delante. En cuanto a la clave foránea, es una columna o un conjunto de columnas en una tabla cuyos valores corresponden a los valores de la clave primaria de otra tabla. Para poder añadir una fila con un valor de clave foránea específico, debe existir una fila en la tabla relacionada con el mismo valor de clave primaria[19].

Por lo mencionado anteriormente, a la hora de crear el modelo y crear las tablas de dimensiones, se ha tenido en consideración que columna es la primaria, es decir, la que indica valores únicos. En este caso se ha visto que ni el producto, ni la marca eran únicos, es más, la combinación de los dos es la que crea productos únicos, ya que hay más de una marca para algunos productos. Por ello se ha creado una nueva columna uniendo la marca y el producto que valdrá como clave primaria. Lo mismo ha ocurrido con las tiendas y las ciudades, ya que en ciertas ciudades hay más de una tienda, por lo que se ha creado una nueva columna con la unión de las dos para crear la maestra y para que sea única. A la hora de crear tanto las tablas de dimensiones como las tablas de hechos se han creado mediante Python. Para crearlas, primero de todo se han seleccionado las columnas relevantes de las tablas que se tienen y se han almacenado en un nuevo *dataframe*. Luego, para los casos que se ha visto que aporta valor, se han añadido nuevas columnas calculadas. Por ejemplo, en la maestra de productos se ha creado una columna llamada categoría donde se agrupan los productos por diferentes rangos de precios. A continuación, se han renombrado las columnas de la tabla ya que estas están en inglés y se les ha dado un nombre claro para todos los casos. En este paso se ha tenido en cuenta que se mantiene la misma nomenclatura en todos los nombres de las columnas. También se han reordenado las columnas para mejorar la legibilidad y se les ha asignado un formato especifico en el caso que fuera necesario. El resultado final es un *DataFrame* que comienza por D (tabla de dimensión) o H (tabla de hechos) estructurado y categorizado, listo para ser utilizado en análisis posteriores. Los *dataframes* mencionados se han exportado al formato Excel para luego insertarlos en el Power BI. En el (Anexo A) se muestran todas las tablas creadas y la descripción de las columnas de cada tabla para tener una visión general de las tablas ingestadas en Power BI.

El anterior paso ha sido el último que se ha dado en Python, por lo que a continuación se ha comenzado con el uso de la herramienta Power BI. Entre varias herramientas existentes para la visualización de los datos se ha seleccionado Microsoft Power BI, que es una colección de servicios de software, aplicaciones y conectores que operan en conjunto para transformar orígenes de datos no relacionados en información coherente, interactiva y visualmente atractiva. Ya sea un simple libro de Microsoft Excel o una colección de almacenes de datos híbridos locales o basados en la nube, permite conectar fácilmente con los orígenes de datos, limpiar y modelar los datos sin afectar al origen subyacente, visualizar lo más relevante y compartirlo con los destinatarios deseados.

Figura 1:Descripción del uso de Power BI para crear análisis controlados por datos

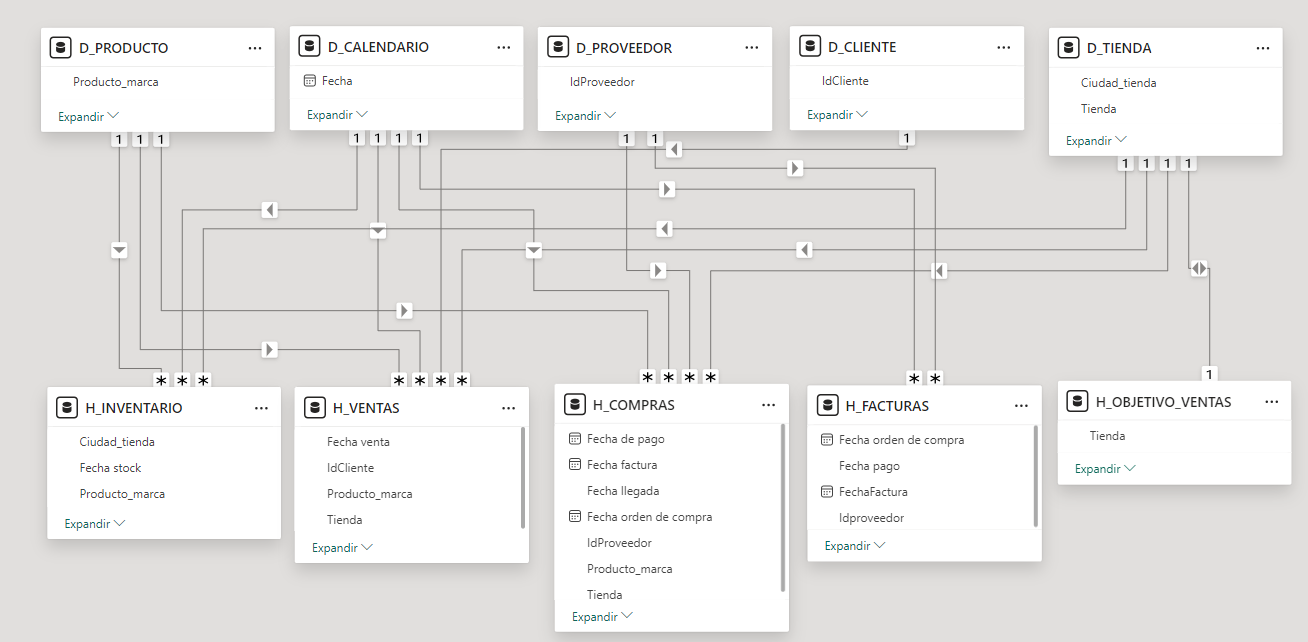
Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Fuente 1:[20]

Al conectar una hoja de cálculo de Excel, es importante que los datos estén en una tabla plana y que cada columna contenga el tipo de datos correcto, como texto, fecha, número o moneda. Además, debe haber una fila de encabezado y no deben existir columnas o filas que muestren totales, ya que las operaciones totales se gestionan en Power BI durante la creación de los objetos visuales[20]. A la hora de realizar la carga de datos al servicio Power BI, este permite crear informes que se conectan a archivos CSV ubicados en el equipo. Por lo que, es así como se ha conectado el conjunto de datos que se ha obtenido tras realizar el procesamiento de ellos en Python. Una vez ingestadas todas las tablas, se ha procedido a crear el modelo semántico. Para ello, se ha tomado en cuenta lo mencionado anteriormente y en la siguiente Figura 3 se observa cómo se ha creado el modelo y las relaciones. En todos los casos excepto en unos la cardinalidad se ha definido uno a varios con la dirección de filtro única.

Figura 3: Diseño del modelo



### Definición de medidas y KPIs

Para la realización de medidas y *Key Performance Indicator (KPI)* se ha utilizado Expresiones de análisis de datos (DAX) que es un lenguaje de expresiones de fórmulas que se usa en *Analysis Services*, Power BI y *Power Pivot* en Excel. Las fórmulas DAX abarcan funciones, operadores y valores para realizar cálculos avanzados y consultas en los datos de las tablas y columnas relacionadas de los modelos de datos tabulares. Las fórmulas DAX se usan en medidas, columnas calculadas, tablas calculadas y seguridad de nivel de fila.

#### Medidas

Las medidas son fórmulas de cálculo dinámico en las que los resultados cambian en función del contexto. Las medidas se usan en informes en los que se pueden combinar y filtrar datos del modelo mediante varios atributos, como un informe de Power BI. Las medidas se crean con la barra de fórmulas DAX del diseñador de modelos. Una fórmula en una medida puede usar las funciones de agregación estándar creadas automáticamente con la característica de Autosuma (como COUNT o SUM), aunque también se puede definir una fórmula propia con la barra de fórmulas DAX.

Al definir una fórmula para una medida en la barra de fórmulas, una característica de información sobre herramientas muestra una vista previa de cuáles serían los resultados para total en el contexto actual, pero de lo contrario no se generan los resultados inmediatamente en ninguna parte. La razón por la que no se pueden ver los resultados (filtrados) del cálculo inmediatamente es que el resultado de una medida no se puede determinar sin el contexto. Se ejecuta una consulta distinta por cada celda de los resultados. Es decir, cada combinación de encabezados de fila y de columna de una tabla dinámica, o cada selección de segmentación de datos y filtros de un informe de Power BI, genera un subconjunto de datos diferente sobre el que se calcula la medida[21]. En este informe se han creado varias medidas, las cuales se han organizado por carpetas para mantener un orden claro de cada una de ellas. En el siguiente (Anexo B) se encuentran las funciones que se han creado.

#### Tablas calculadas

Una tabla calculada es un objeto calculado, basado en una expresión de fórmula, que se deriva de todas las tablas (o parte de ellas) del mismo modelo. En lugar de consultar y cargar valores en las columnas de la nueva tabla desde un origen de datos, una fórmula DAX define los valores de la tabla. Las tablas calculadas admiten relaciones con otras tablas, tienen tipos de datos y formato, y pueden pertenecer a una categoría de datos[21].

La tabla D\_CALENDARIO donde se registran las fechas para después relacionarla con las tablas de hecho, se ha creado utilizando DAX, mediante la función CALENDAR(). Esta función devuelve un intervalo de fechas contiguo en función de fechas de inicio y de finalización que se especifican como argumentos en la función. La fecha de inicio se elige como la más temprana del modelo semántico y la fecha de finalización es la última[22]. En este caso se han cogido los datos desde 2015 a 2017 ya que el análisis se basa en los datos de 2016. Una vez creada la tabla se cuenta con una columna de fechas que se puede usar. A través de la columna fecha, utilizando las funciones de DAX, se han obtenido otras columnas como la del año, el número del mes, la semana del año y el día de la semana.

## Visualización

Una vez conectados los datos, diseñado el modelo y creado las medidas necesarias, se ha comenzado a crear el informe. En el lado derecho se encuentran los paneles de visualizaciones, filtros y campos. Los datos de la tabla del libro de Excel aparecen en el panel campos, mostrando el nombre de la tabla, en la parte superior, y los encabezados de columna como campos individuales debajo.

### Estructura del informe

Estructuralmente, el informe de Power BI se conecta a un único modelo semántico y tiene varias páginas de informe. En cada página, se han diseñado objetos de informe, entre los que se incluyen los siguientes: Objetos visuales, que serían visualizaciones de datos del modelo semántico y elementos, que proporcionan interés visual, pero no usan datos del modelo semántico. Entre los elementos se incluyen cuadros de texto, botones, formas e imágenes.

### Especificación del diseño del informe analítico

Tener datos correctos y seleccionar los objetos visuales adecuados es crucial, pero también es importante que el informe sea visualmente atractivo. Un buen diseño debe guiar al consumidor para encontrar y comprender rápidamente las respuestas a sus preguntas. El informe no solo debe ser atractivo, sino también fácil de usar. Al observar un informe, los consumidores pasan por un proceso automático e inconsciente para comprender lo que ven. Por ello, es fundamental cumplir con los principios básicos de diseño de informes para comunicar eficazmente el significado de los datos, combinando ciencia y arte. El diseño del informe comienza determinando el número, secuencia y propósito de las páginas, evitando combinar temas u objetivos opuestos en la misma página. Luego, se especifica cada diseño de página con los objetos de informe pertinentes[23].

#### Espacio, márgenes y equilibrio

El espacio es esencial para un diseño eficaz, ya que ayuda a reducir el desorden y aumentar la legibilidad. Este se aplica a los márgenes de la página del informe, es decir, el área de borde que rodea cada página, y entre los objetos del informe cuando consta de varios grupos de objetos relacionados. Los objetos de informe deben estar enmarcados por un área de borde con un espaciado coherente y los márgenes deben ser iguales a la izquierda y a la derecha, pero pueden variar en la parte superior e inferior. El equilibrio, también, en la disposición de objetos es fundamental para la estabilidad y la estructura del diseño de un informe, refiriéndose al peso distribuido en la página mediante la colocación de objetos de tamaños iguales o diferentes. En este caso se ha jugado con un equilibrio asimétrico.

#### Tamaño del objeto visual

#### Alineación y tamaño del objeto visual

Por un lado, cuando haya varios objetos visuales en la página del informe, deben estar correctamente alineados, es decir, los bordes de los objetos visuales y los espaciados entre los objetos visuales deben ser coherentes. Además, generalmente, se debe ubicar la información más importante en la esquina superior izquierda de la página y organizar los elementos de izquierda a derecha y de arriba abajo. Los objetos relacionados deben agruparse lógicamente, lo que crea una conexión visual y evita el desorden. Además, la alineación visualmente atractiva de los objetos puede transmitir más energía e interés que simplemente centrar o colocar aleatoriamente los objetos de informe. Por otro lado, el tamaño de un objeto visual indica su importancia ya que los consumidores de informes suelen enfocarse primero en los objetos más grandes.

#### Color

El color debe emplearse con moderación y propósito, evitando su uso excesivo para no distraer al lector. Se han utilizado unos pocos colores suaves, alineados con la paleta corporativa. Esto asegura que los datos sean el foco principal del informe. Los colores llamativos deben se han reservado para resaltar excepciones y el contraste se ha usado para combinar dos objetos opuestos.

#### Coherencia en el diseño de informes

Es fundamental mantener la coherencia al diseñar y configurar objetos de informe. Esta coherencia debe reflejarse en todos los aspectos del diseño del informe, incluyendo el espaciado, los márgenes, el tamaño, la alineación y, especialmente, las opciones de formato del objeto. Un buen uso de la repetición también crea asociación y coherencia y ayuda a reforzar el diseño de un informe asociando objetos de informe relacionados. En el informe se ha utilizado el mismo diseño para analizar las compras y las ventas.

#### Selección de objetos visuales en informes

El objetivo principal de la visualización de datos es comunicar información de manera clara y efectiva a los consumidores del informe. Por ello, es crucial seleccionar el tipo de objeto visual más adecuado para cumplir con los requisitos específicos.

Los gráficos de barras o columnas se han usado para datos que contienen varias categorías. Y en caso de tener una serie temporal se ha utilizado el gráfico de líneas para mostrar valores en el tiempo, presentando la fecha en el eje X ordenado de los períodos más antiguos a los más recientes. Los objetos visuales proporcionales representan los datos como partes de un todo, comunicando eficazmente la distribución de un valor en una dimensión. Los gráficos de columnas y barras son especialmente efectivos para visualizar proporciones en múltiples dimensiones. Los valores numéricos, son representados mediante objetos visuales tipo tarjeta, ya que destacan información crucial que requiere atención inmediata. Son efectivos en informes analíticos y de panel, ya que comunican datos importantes de manera rápida y directa.

Las tablas y matrices son recursos subestimados que pueden comunicar eficazmente una gran cantidad de información detallada. Las tablas tienen un número fijo de columnas que pueden mostrar datos agrupados o resumidos. Por otro lado, las matrices permiten organizar datos en grupos tanto en columnas como en filas. La aplicación de opciones de formato condicional, como colores de fondo, colores de fuente o iconos, puede realzar visualmente los valores con indicadores visuales, facilitando así la interpretación de informes y equilibrando la presentación en la página. Además, las matrices ofrecen una excelente experiencia para la navegación jerárquica, permitiendo a los usuarios explorar profundamente en columnas y filas para descubrir datos detallados de interés. Las opciones de formato disponibles para tablas y matrices ofrecen un alto grado de control sobre el estilo y la presentación de los datos de cuadrícula. Por último, cuando un modelo semántico contiene información geoespacial, esta puede visualizarse eficazmente mediante objetos visuales de mapa. Power BI ofrece varios objetos visuales principales diseñados para mapas, cada uno con diversas opciones de formato que, aplicadas correctamente, pueden resaltar datos geoespaciales de manera efectiva.

También es importante la selección de objetos visuales adaptados al diseño del informe. Al elegir entre diversos tipos de objetos visuales para cumplir con los requisitos de diseño, es útil considerar el espacio disponible en la página del informe. Seleccione un objeto visual que sea estéticamente atractivo y, al mismo tiempo, maximice la utilización del espacio disponible en la página[24].

#### Formato y configuración de visualizaciones

Power BI Desktop proporciona diversas opciones para personalizar el aspecto de las visualizaciones seleccionadas, como los colores y el formato del texto. Las opciones de formato disponibles varían según el tipo de visualización seleccionada. Entre las más comunes se encuentran Título, Fondo y Borde.

* En la sección título, se agrega o edita el título de la visualización para describir claramente los datos presentados. Se puede ajustar el tamaño del texto, la fuente, el color y la alineación del título.
* En la sección fondo, se establece un color de fondo para la visualización. Es recomendable utilizar un fondo claro para asegurar que los datos sean legibles.
* La sección borde permite definir un borde alrededor del objeto visual para destacarlo en el lienzo y mejorar su claridad visual, en el informe se ha optado por poner el borde redondeado.

En el panel formato, también se ajustan los colores y las etiquetas de valores de datos específicos. En la sección colores de datos, se configuran los colores que representan diferentes campos de datos dentro de la visualización. Se ha mantenido consistencia en la elección de colores para la coherencia del informe. También existe la opción información sobre herramientas que permite agregar detalles adicionales que aparecen al mantener el ratón sobre la visualización[24].

#### Trabajo con marcadores

Los marcadores ofrecen una forma efectiva de transformar un informe en una experiencia analítica guiada, maximizando el espacio de página y facilitando interacciones intuitivas. Pueden variar desde acciones simples, como restablecer filtros, hasta comportamientos complejos que alteran objetos visuales o exploran en profundidad. En el informe se ha hecho uso de marcadores para cambiar la visualización de gráfica a tabla. Para ello, se han añadido botones y se han configurado sus acciones.

A continuación, se ha explicado brevemente como se han configurado los marcadores definiendo los estados que capturan y el ámbito de los objetos visuales afectados. Por defecto, un nuevo marcador captura todos los tipos de estado, pero se puede optar por deshabilitar cualquiera de ellos según sea necesario [24].

* Estado Datos: Captura las configuraciones de consulta que afectan al modelo semántico, como segmentaciones y criterios de ordenación. También incluye la profundidad de los detalles de los objetos visuales afectados por la consulta.
* Estado Mostrar: Se refiere a la visibilidad de los objetos en el informe, incluyendo objetos visuales, cuadros de texto, botones, formas e imágenes. Permite ocultar o mostrar objetos según el marcador aplicado.
* Estado Página actual: Decide si el marcador dirige al usuario a la página marcada o si aplica la página actual.

El ámbito del marcador determina a qué objetos visuales se aplica:

* Todos los objetos visuales: Aplica el marcador a todos los objetos del informe, incluso aquellos que están ocultos.
* Objetos visuales seleccionados: Limita el marcador a los objetos visuales específicamente seleccionados al momento de configurar el marcador.

## Aplicación de Modelos para la predicción de la demanda

Después de revisar varios modelos de *ML* en el marco teórico, se ha procedido a aplicar un ejemplo de modelo simple para la licorería siguiendo un enfoque práctico. Para ilustrar este proceso, se ha seleccionado un modelo de los previamente mencionados y se ha implementado para predecir la demanda utilizando datos históricos disponibles.

A la hora de crear el modelo de predicción, primero de todo, se ha realizado la consolidación de datos, donde se han combinado los datos de ventas y compras en un único *dataset* para facilitar el análisis. En ambos casos, se han cogido los datos desde el 01-01-2016 hasta el 29-02-2016. Luego se ha realizado un Análisis Exploratorio de Datos (EDA) para entender mejor las características de los datos y detectar p patrones, tendencias y posibles anomalías. En este caso se ha visualizado la serie temporal de ventas y compras, como se ve en la siguiente Figura 2.

Figura 4:Serie temporal de ventas y compras

Gráfico, Histograma

Descripción generada automáticamente

A continuación, se han generado nuevas características que puedan ser útiles para el modelo de predicción, como tasas de ventas diarias, ventas y compras promedio por semana/mes etc. Luego se ha seleccionado el modelo de predicción, donde al principio se ha realizado un modelo de regresión, pero luego se ha aplicado un modelo de series temporales viendo que puede tener patrones estacionales. Seguido se ha entrenado el modelo utilizando los datos disponibles y se ha validado su desempeño. En esta fase, primero se han dividido los datos en conjuntos de entrenamiento y validación. Y luego se ha entrenado el modelo seleccionado con los datos de entrenamientos. Por último, se ha evaluado el modelo utilizando métricas de evaluación las cuales han sido MAE (*Mean Absolute Error*) y RMSE (*Root Mean Square Error*). La métrica de evaluación MAE (Mean Absolute Error o Error Absoluto Medio) es una medida de la precisión de un modelo de predicción. Representa la media de las diferencias absolutas entre las predicciones del modelo y los valores reales observados. Un valor de MAE más bajo indica un modelo más preciso. En este caso, tras realizar el modelo de regresión lineal, el valor MAE: 224483.14. Esto significa que, en promedio, las predicciones del modelo están desviadas de los valores reales por aproximadamente 224483.14 unidades de la cantidad. El RMSE es una medida comúnmente utilizada para evaluar la precisión de un modelo de regresión o de predicción. Se calcula como la raíz cuadrada de la media de los errores al cuadrado entre los valores predichos por el modelo y los valores observados reales. Este mide la diferencia promedio entre los valores predichos y los valores reales en la misma escala que los datos originales. Cuanto más bajo sea el valor de RMSE, mejor será la capacidad predictiva del modelo. En este caso el valor obtenido es 224578.26. Al tener una escala de datos muy grande, es decir, los datos de ventas y compras son de gran magnitud, puede llegar a ser aceptable pero aun así se ha intentado mejorar realizando otro modelo de predicción más avanzado como ARIMA, SARIMA, o modelos de aprendizaje automático como *Random Forest*, *Gradient Boosting*, etc.

Después de encontrar resultados subóptimos con un enfoque de regresión lineal para predecir la demanda, se ha decidido explorar la eficacia del modelo de *Random Forest*. Este modelo más avanzado ofrece la capacidad de manejar relaciones no lineales y capturar la complejidad de los datos de manera más efectiva. En este caso, primero de todo se han importado las bibliotecas necesarias: *RandomForestRegressor* para el modelo de regresión forestal aleatoria, *GridSearchCV* para la búsqueda de hiperparámetros, y *train\_test\_split* para dividir los datos en conjuntos de entrenamiento y prueba. A continuación, se han definido las características (x) y la variable objetivo (y) para el modelo y se ha dividido el conjunto de datos en conjuntos de entrenamiento y prueba con una proporción del 20% de los datos reservados para pruebas y un estado aleatorio fijado en 42. Para obtener el resultado más optimo, se ha definido un diccionario de hiperparámetros para probar diferentes combinaciones de valores y se ha realizado una búsqueda con validación cruzada de 3 pliegues y métrica de evaluación configurada en el error absoluto medio negativo. Después, se ha ajustado el modelo con los mejores hiperparámetros encontrados durante la búsqueda y se han realizado las predicciones en el conjunto de prueba utilizando el mejor modelo. Y por último se han calculado el error absoluto medio (*MAE*) y el error cuadrático medio (*RMSE*) entre las predicciones y los valores reales en el conjunto de prueba. Se imprimen los resultados de *MAE* y *RMSE*. Estos valores indican la diferencia promedio absoluta y la raíz del error cuadrático medio entre las predicciones del modelo y los valores reales en el conjunto de prueba, respectivamente. Los resultados del modelo *Random Forest* muestran una mejora significativa en comparación con el enfoque de regresión lineal anterior. El error absoluto medio (*MAE*) y el error cuadrático medio de la raíz (*RMSE*) indican que las predicciones del modelo tienen un desempeño prometedor, con un MAE de aproximadamente 60981 unidades y un RMSE de alrededor de 69531 unidades. Estas métricas son esenciales para evaluar la precisión del modelo, y valores más bajos indican una mejor capacidad de predicción. Además, los coeficientes de determinación, tanto en el conjunto de entrenamiento (0.889) como en el conjunto de prueba (0.914), son altos. Esto sugiere que una gran parte de la variabilidad en la variable dependiente puede ser explicada por las características incluidas en el modelo. Un coeficiente de determinación cercano a 1 indica un buen ajuste del modelo a los datos, lo que confirma la capacidad predictiva del modelo *Random Forest*. En resumen, estos resultados indican que el modelo *Random Forest* ha logrado capturar de manera efectiva la relación entre las variables predictoras y la variable objetivo, superando notablemente el desempeño del modelo de regresión lineal anterior. Esto respalda la elección del modelo *Random Forest* como un enfoque más adecuado para la tarea de predicción en este contexto específico.

En este estudio, además de explorar el rendimiento del modelo, también se ha considerado la aplicación de un enfoque alternativo utilizando un modelo ARIMA. Sin embargo, antes de proceder con el modelo ARIMA, se ha llevado a cabo un análisis para determinar si la serie temporal subyacente es estacionaria o no. Esto implica que la media y la varianza de la serie no cambian con el tiempo. Este paso es crucial, ya que los modelos ARIMA requieren que la serie sea estacionaria para producir predicciones precisas. En este sentido, se ha investigado la estacionariedad de la serie temporal a través de la Prueba de *Dickey-Fuller* aumentada (ADF test). Este análisis proporciona información fundamental sobre la naturaleza de los datos y orienta la selección del modelo más apropiado para su modelado.

Figura 5: Ventas diarias

Gráfico, Histograma

Descripción generada automáticamente

La prueba de Dickey-Fuller aumentada (ADF) se ha utilizado para determinar si la serie temporal es estacionaria o no. La estadística ADF obtenida es -0.6876. Para decidir sobre la estacionariedad, se ha comparado esta estadística con los valores críticos correspondientes a los niveles de significancia del 1%, 5% y 10%. El valor p de la prueba es 0.8499. Un valor p alto (mayor que el nivel de significancia comúnmente usado como el 5%) indica que no se puede rechazar la hipótesis nula, lo que sugiere que la serie no es estacionaria. En resumen, dado que la estadística ADF es mayor que los valores críticos en todos los niveles de significancia y el valor p es significativamente alto, se puede concluir que la serie temporal no es estacionaria.

Cuando la serie temporal no es estacionaria, es necesario realizar algunas transformaciones para convertirla en estacionaria antes de poder aplicar modelos de series temporales como ARIMA de manera efectiva. Entre las estrategias más comunes existentes, se ha aplicado la de remover la estacionalidad. Esto se ha realizado restando la media móvil estacional o tomando la diferencia entre una observación y su valor en el mismo punto en el ciclo estacional anterior.

En el código, se ha calculado la media móvil estacional con una ventana de 7 días. Luego, se ha restado la media móvil estacional de la serie temporal original para eliminar la estacionalidad y obtener una nueva serie temporal. Finalmente, se ha visualizado tanto la serie temporal original como la serie sin estacionalidad para comparar cómo se ven en la Figura 4.

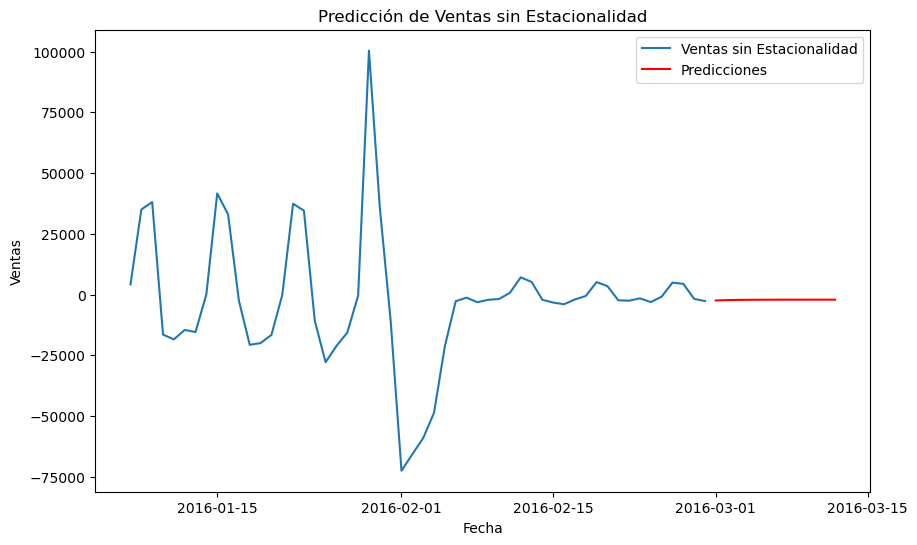
Figura 6: Ventas diarias con y sin estacionalidad

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Después de eliminar la estacionalidad, se ha procedido a ajustar un modelo ARIMA a la serie temporal para realizar predicciones. En el código, se ha ajustado un modelo ARIMA a la serie temporal sin estacionalidad utilizando la clase ARIMA de *statsmodels*. Después de entrenar el modelo, se han hecho las predicciones para los próximos 12 pasos adelante y se han visualizado en la Figura 5 dichas predicciones junto con la serie temporal original.

Figura 7: Predicción de ventas sin estacionalidad



## Control de versiones en GIT

Git es un sistema avanzado y distribuidor de control de versiones que permite rastrear el progreso de un proyecto a lo largo del tiempo mediante capturas del mismo mientras evoluciona, registrando quién ha hecho qué cambios y por qué. Además, facilita el trabajo en paralelo de varios participantes, permitiendo que todos trabajen en el proyecto simultáneamente. GitHub es un servidor de alojamiento en línea para proyectos basados en Git, permitiendo la colaboración entre usuarios. Funciona como un repositorio remoto donde se almacenan los archivos de un proyecto y registra su desarrollo de manera remota. Facilita compartir proyectos, registrar el desarrollo y proporciona seguridad en la nube. En proyectos colaborativos, GitHub se considera la copia principal del proyecto, siendo el punto central de interacción entre Git y los colaboradores[25].

Aunque el trabajo colaborativo no haya sido relevante en este caso específico, el uso de Git ha sido beneficioso para registrar y gestionar de manera efectiva las versiones del proyecto. Ha ayudado a mantener un flujo de trabajo organizado, a tener un registro claro de los cambios realizados y a facilitar futuras actualizaciones y mejoras en el proyecto. A continuación, se mencionan lo beneficios de manera más detallada.

En primer lugar, Git proporciona un historial detallado de todos los cambios realizados en el código a lo largo del tiempo. Esto es invaluable para entender cómo ha evolucionado el proyecto, qué decisiones se tomaron en cada paso del camino y por qué se tomaron. Incluso para un desarrollador individual, tener esta información a mano puede ser útil para rastrear errores, probar diferentes enfoques y mejorar la eficiencia en el desarrollo.

Además, Git ofrece la capacidad de crear ramas para experimentar con nuevas características o realizar cambios importantes sin afectar el código principal. Esto permite una mayor libertad y flexibilidad en el proceso de desarrollo, ya que los cambios pueden probarse y revisarse antes de fusionarse con la rama principal del proyecto.

Otro beneficio es la posibilidad de revertir a versiones anteriores en caso de que surjan problemas o se necesite retroceder a una versión estable anterior. Esto proporciona una red de seguridad invaluable y tranquilidad al saber que es posible recuperar el proyecto en caso de emergencia.

La integración efectiva de GitHub en *Visual Studio Code* comienza con la configuración y autenticación, garantizando una conexión segura entre el entorno de desarrollo local y los repositorios en GitHub. Para empezar, es necesario tener instalado *Visual Studio Code*, y disponer de una cuenta de GitHub. Para vincular la cuenta de GitHub en *Visual Studio Code*, se ha autentificado mediante la generación de un token de acceso personal en GitHub, el cual se introduce en *Visual Studio Code*.

Una vez configurada y autentificada ambas cuentas, se han realizado las operaciones básicas como la creación de un nuevo repositorio en GitHub y la clonación de repositorios existentes directamente desde la interfaz de Visual Studio Code. También se han gestionado cambios utilizando la interfaz gráfica de Visual Studio Code, incluyendo el *staging* y *commit* de cambios, así como el *push* y *pull* de cambios hacia y desde el repositorio remoto en GitHub[26].

# RESULTADOS

Tras un exhaustivo proceso de limpieza y transformación de datos, se ha logrado desarrollar un informe interactivo que cumple con los objetivos establecidos para el proyecto. El proceso de limpieza de datos ha sido fundamental para garantizar la calidad y precisión de la información, eliminando inconsistencias, datos duplicados, y valores atípicos que podrían afectar el análisis final. El resultado final es un informe interactivo que no solo presenta los datos de manera clara y comprensible, sino que también permite al cliente interactuar con la información de una manera que se alinea con sus necesidades y objetivos específicos para la toma de decisiones. Este enfoque interactivo no solo mejora la comprensión de los datos, sino que también facilita una toma de decisiones más informada y eficiente.

Explicar la funcionalidad del informe de PowerBI y como se interactúa entre páginas

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Mapa

Descripción generada automáticamente

# CONCLUSIONES Y DISCUSIÓN

xxxxxxxx

**BIBLIOGRAFÍA**

[1] Jordi de Mas Jaumot, “Gestión del inventario en una empresa del sector farmacéutico mediante algoritmos de Machine Learning,” 2021. [Online]. Available: https://openaccess.uoc.edu/bitstream/10609/129826/8/jde\_masTFM0121memoria.pdf

[2] I. de I. Conocimiento, “Machine Learning & Deep Learning,” 2024, [Online]. Available: https://www.iic.uam.es/inteligencia-artificial/machine-learning-deep-learning/

[3] T. Šustrová, “A Suitable Artificial Intelligence Model for Inventory Level Optimization Tereza Šustrová: A Suitable Artificial Intelligence Model for Inventory Level Optimization,” vol. 25, no. 1, pp. 48–55, 2016, [Online]. Available: http://dx.doi.org/10.13164/trends.2016.25.48

[4] S. Zhang, X. Qin, S. Hu, Q. Zhang, B. Dong, and J. Zhao, “Importance Degree Evaluation of Spare Parts Based on Clustering Algorithm and Back-Propagation Neural Network,” *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/6161825.

[5] K. B. Praveen, P. Kumar, J. Prateek, G. Pragathi, and P. M. J, “Inventory Management System Using Machine Learning,” *Int. J. Innov. Eng. Manag. Res.*, vol. 9, no. 06, pp. 769–785, 2022, doi: 10.48047/ijiemr/v11/i06/51.

[6] Scikit learn, “Gradient Boosting Regressor,” 2024. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

[7] “Decision tree in Machine Learning,” 2024, [Online]. Available: https://www.geeksforgeeks.org/decision-tree-introduction-example/

[8] Al Maverick, “Light Gradient Boosting Machine,” 2023. https://samanemami.medium.com/light-gradient-boosting-machine-b4f1b9e3f7d1

[9] Gonzalo Castillo, “Logística 4.0,” *Innovación Digit. 360*, 2023, [Online]. Available: https://www.innovaciondigital360.com/industria-4-0/supply-chain/logistica-4-0-que-es-caracteristicas-beneficios-y-tecnologia/

[10] EAE Business School, “Descubre la Optimización de Inventario con Machine Learning,” 2024. https://www.eaemadrid.com/es/blog/machine-learning

[11] Kaggle, “Inventory Analysis case study,” 2023. https://www.kaggle.com/datasets/bhanupratapbiswas/inventory-analysis-case-study

[12] P. D. A. Library, “Pandas,” 2024. https://pandas.pydata.org/

[13] “Numpy,” 2024. https://numpy.org/

[14] Matplotlib, “Visualization with python,” 2024. https://matplotlib.org/

[15] Seaborn, “Seaborn: Statistical data visualization,” 2024, [Online]. Available: https://seaborn.pydata.org/

[16] Microsoft, “Diseño de un modelo semántico en PowerBI,” 2023. https://learn.microsoft.com/es-es/training/modules/design-model-power-bi/1-introduction

[17] J. Quiroz, “El modelo relacional de bases de datos,” *Boletín de Política Informática*, vol. 6, pp. 53–61, 2003.

[18] Microsoft, “Crear y administras relaciones en PowerBI,” 2024, [Online]. Available: https://learn.microsoft.com/es-es/power-bi/transform-model/desktop-create-and-manage-relationships

[19] IBM, “Primary and foreign keys,” 2023, [Online]. Available: https://www.ibm.com/docs/es/ida/9.1.2?topic=entities-primary-foreign-keys

[20] Microsoft learn, “Descripción del uso de Power BI para crear análisis controlados por datos,” 2024. [Online]. Available: https://learn.microsoft.com/es-es/training/modules/introduction-power-bi/2-describe-using-power-bi-build-data-driven-analytics

[21] Microsoft learn, “Información general sobre DAX,” 2024, [Online]. Available: https://learn.microsoft.com/es-es/dax/dax-overview

[22] Microsoft, “Creación de una tabla de fechas,” 2024, [Online]. Available: https://learn.microsoft.com/es-es/training/modules/design-model-power-bi/3-date-table

[23] Microsoft learn, “Diseño de informes en Power BI,” 2024, [Online]. Available: https://learn.microsoft.com/es-es/training/modules/power-bi-effective-reports/

[24] “Mejora de los diseños de informes de Power BI para la experiencia del usuario,” 2024, [Online]. Available: https://learn.microsoft.com/es-es/training/modules/power-bi-effective-user-experience/

[25] J. Astigarraga and V. Cruz-Alonso, “¡Se puede entender cómo funcionan Git y GitHub!,” *Ecosistemas*, vol. 31, no. 1, pp. 1–6, 2022, doi: 10.7818/ECOS.2332.

[26] Github, “Uso de GitHub Codespaces en Visual Studio Code,” 2024. https://docs.github.com/es/codespaces/developing-in-a-codespace/using-github-codespaces-in-visual-studio-code

|  |
| --- |
| Anexos / Eranskinak |

Anexo A: Descripción de las tablas del modelo

|  |  |  |
| --- | --- | --- |
| Tabla | Columna | Descripción |
| D\_PRODUCTO | Producto\_marca | Clave primaria. Es un denominador único de la tabla que se utiliza para relacionar con la tabla de hechos. |
| Marca | Marca del producto |
| Producto | Nombre del producto |
| Precio compra unidad | El precio de compra de cada producto |
| Precio venta unidad | El precio de venta de cada producto |
| Margen bruto | El margen bruto por cada producto, es decir la resta entre el precio de venta y compra |
| Volumen | El volumen en litros de cada producto |
| Categoría | Se han dividido 5 categorías (Low cost, economic, intermediate, Premium y luxury) dependiendo del precio unidad de cada producto |
| D\_CLIENTE | IdCliente | Clave primaria. Es un denominador único de la tabla que se utiliza para relacionar con la tabla de hechos |
| Cliente | Nombre del cliente |
| D\_PROVEEDOR | IdProveedor | Clave primaria. Es un denominador único de la tabla que se utiliza para relacionar con la tabla de hechos |
| Proveedor | Nombre del proveedor |
| D\_TIENDA | Ciudad\_tienda | Clave primaria. Es un denominador único de la tabla, que se utiliza para relacionar con la tabla de hechos |
| Tienda | Nombre de la tienda |
| Ciudad | Nombre de la ciudad en la que se encuentra la tienda, ya que hay una única tienda por ciudad |
| Estado | El nombre del estado de la ciudad donde se encuentra la tienda |
| D\_CALENDARIO | Fecha | Clave primaria. Fecha completa construida mediante la función CALENDAR() |
| Año | Año de la fecha |
| NumMes | Número del mes de la fecha |
| NumSemana | Número de la semana de la fecha indicada |
| Día de la semana | Día de la semana correspondiente |
| Mes | Mes correspondiente |
| NumDía | Número del día correspondiente del mes |
| H\_VENTAS | Tienda | Clave foránea. Código de la tienda que se relaciona con la tabla M\_TIENDA |
| Producto\_marca | Clave foránea. Código del producto que se relaciona con la tabla M\_PRODUCTO |
| IdCliente | Clave foránea. Código del cliente que se relaciona con la tabla M\_CLIENTE |
| Fecha venta | Fecha de la venta de producto |
| Unidades vendidas | Cantidad de ventas realizadas |
| Precio venta unidad | Precio unitario del producto vendido |
| Precio sin impuesto | Precio de venta considerando la cantidad vendida, pero sin tener en cuenta el impuesto |
| Impuesto | Precio del impuesto |
| Ventas | Columna calculada. Precio total considerando la cantidad y el impuesto |
| H\_COMPRAS | Tienda | Clave foránea. Código de la tienda que se relaciona con la tabla D\_TIENDA |
| Producto\_marca | Clave foránea. Código del producto que se relaciona con la tabla D\_PRODUCTO |
| IdProveedor | Clave foránea. Código del proveedor que se relaciona con la tabla D\_PROVEEDOR |
| Fecha orden de compra | Fecha de cuando se realizó la compra |
| Fecha llegada | Clave foránea. Fecha de la llegada del producto a planta, que se relaciona con D\_CALENDARIO |
| Fecha factura | Fecha de la factura |
| Fecha de pago | Fecha de pago |
| Precio compra unidad | El precio unitario de compra |
| Unidades compradas | La cantidad de compra |
| Coste compras | El precio total de la compra, considerando la cantidad |
| Plazo de entrega | Columna calculada, indica el plazo desde que se realizó la orden de compra hasta que llega el producto. |
| H\_INVENTARIO | Ciudad\_tienda | Clave foránea. Código de la tienda que se relaciona con la tabla D\_TIENDA |
| Producto\_marca | Clave foránea. Código del producto que se relaciona con la tabla D\_PRODUCTO |
| Stock | Cantidad de stock el día 01/01/2016 |
| Fecha stock | Fecha cuando se contabilizó el stock |
| Estado | Estado donde se almacena el stock |
| H\_FACTURAS | IdProveedor | Clave foránea. Código del proveedor que se relaciona con la tabla D\_PROVEEDOR |
| Fecha factura | Fecha de llegada de la factura |
| Numero orden de compra | Número de la orden de compra |
| Fecha orden de compra | Fecha de cuando se emitió la orden de compra |
| Fecha pago | Fecha de cuando se pago |
| Unidades | Unidades vendidas |
| Coste | Coste de la compra |
| Transporte | Coste del transporte |

Anexo B: Descripción de las medidas calculadas

|  |  |  |
| --- | --- | --- |
| Carpeta | Nombre | Descripción |
| Compras | Cantidad compras | Suma de la cantidad de compras registrada |
| Máximo plazo de entrega | La máxima cantidad de días para entregar un producto desde que se ha realizado el pedido |
| Mínimo plazo de entrega | La mínima cantidad de días para entregar un producto desde que se ha realizado el pedido |
| Media plazo de entrega | La media de días para entregar un producto desde que se ha realizado el pedido |
| Precio compra total | La suma del precio de compra total |
| Precio compra unidad | La suma del precio de compra por unidad |
| Transporte | Coste del transporte |
| Stock | Stock optimo | Cálculo del stock óptimo |
| Stock seguridad | Cálculo del stock de seguridad |
| Ventas | Cantidad ventas | Suma de la cantidad de ventas registrada |
| Cantidad ventas enero | Suma de la cantidad de ventas registrada en el mes de enero |
| Días con ventas | Cantidad de días que se han registrado ventas |
| Impuesto | Suma del impuesto por venta de alcohol |
| Máximo ventas | La máxima cantidad productos que se han vendido en un mismo pedido |
| Mínimo ventas | La mínima cantidad productos que se han vendido en un mismo pedido |
| Media ventas | La cantidad de productos que se venden de media en un mismo pedido |
| Precio ventas total | La suma del precio de compra total |
| Precio ventas unidad | La suma del precio de compra por unidad |
| Ventas diarias | Cálculo de la división entre la cantidad de ventas en total y el número de días con ventas |
| Utilidad | Ciudad seleccionada | Cálculo que indica la ciudad que ha seleccionado el usuario en el filtro |
| Cliente seleccionado | Cálculo que indica el cliente que ha seleccionado el usuario en el filtro |
| Producto seleccionado | Cálculo que indica el producto que ha seleccionado el usuario en el filtro |
| Proveedor seleccionado | Cálculo que indica el proveedor que ha seleccionado el usuario en el filtro |
| Fecha seleccionada | Cálculo que indica la fecha que ha seleccionado el usuario en el filtro |
| Título del gráfico | Cálculo que muestra el título que tu configuras en el gráfico |

