

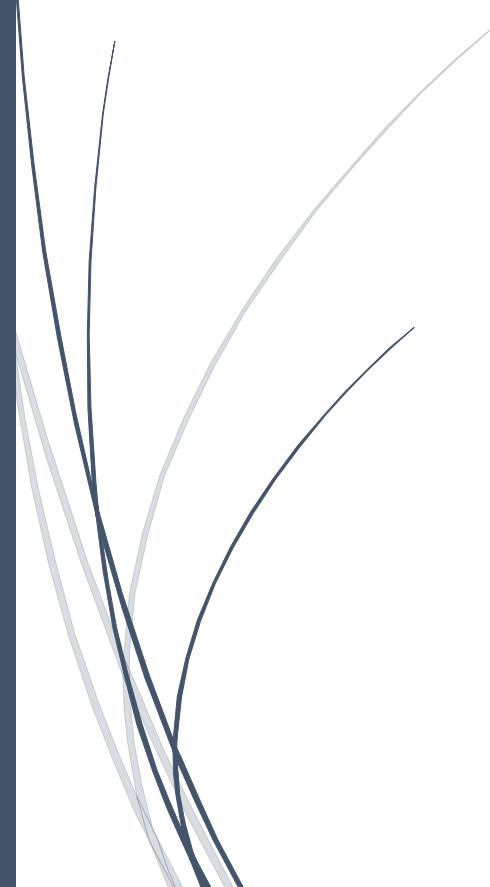


FEBRERO 2026

OCANTOS, GISELA AYELEN

# PROYECTO FINAL SQL

CODER HOUSE – COMISIÓN 81820



## SISTEMA DE GESTIÓN DE STOCK

### INTRODUCCIÓN:

Este documento presenta una base de datos relacional para un sistema de gestión de ventas, compras y stock.

El objetivo principal es modelar la información necesaria para una empresa que comercializa productos, integrando áreas como ventas, logística, análisis comercial y abastecimiento.

### OBJETIVO:

El presente proyecto tiene como objetivo desarrollar una base de datos para un sistema de ventas que permita a una empresa gestionar de manera eficiente sus operaciones comerciales, incluyendo el registro de clientes, proveedores, ventas y compras. Asimismo, el sistema busca facilitar el control y seguimiento del stock disponible en depósitos, detectar productos en riesgo de reposición y generar información relevante para la toma de decisiones comerciales.

Implementando el siguiente programa del ciclo comercial

- Gestión de clientes
- Registro de ventas
- Control de detalle de venta (producto–precio–cantidad)
- Gestión de vendedores y zonas
- Métodos de pago
- Envíos
- Gestión de proveedores
- Compras y detalle de compra
- Control de productos
- Stock mínimo y alertas de reposición
- Movimientos de stock (entrada/salida)
- Listas de precios y porcentajes de ganancia
- Depósitos de almacenamiento
- Clasificación por categorías

### SITUACIÓN PROBLEMÁTICA:

Muchas pequeñas y medianas empresas utilizan para gestionar sus procesos muchas hojas de cálculo separadas sin relación alguna, controles deficientes de stock, precios y dificultades para determinar el momento de comprar su mercadería, todo esto conlleva a generar pérdidas económicas y a no poder obtener reportes confiables para poder tomar buenas decisiones que ayuden al crecimiento del negocio.

**LISTADO DE TABLAS:**

CLIENTE	Tipo de dato	PK	FK	Referencia	Descripción
cliente_id	INT	✓			Identificador del cliente
apellido	VARCHAR(100)				Apellido del cliente
nombre	VARCHAR(100)				Nombre del cliente
genero	ENUM				Género del cliente
direccion	VARCHAR(150)				Dirección del cliente
email	VARCHAR(100)				Correo electrónico
fecha_nacimiento	DATE				Fecha de nacimiento
fecha_alta	DATE				Fecha de alta
estado	VARCHAR(50)				Estado del cliente

METODO PAGO	Tipo de dato	PK	FK	Referencia	Descripción
metodo_pago_id	INT	✓			Identificador del método
tipo	VARCHAR(50)				Tipo de pago

ZONA	Tipo de dato	PK	FK	Referencia	Descripción
zona_id	INT	✓			Identificador de zona
sector	VARCHAR(50)				Sector de venta

PROVEEDOR	Tipo de dato	PK	FK	Referencia	Descripción
proveedor_id	INT	✓			Identificador del proveedor
razon_social	VARCHAR(150)				Razón social
direccion	VARCHAR(150)				Dirección
pais	VARCHAR(50)				País de origen

DEPOSITO	Tipo de dato	PK	FK	Referencia	Descripción
deposito_id	INT	✓			Identificador del depósito
direccion	VARCHAR(150)				Dirección del depósito
tipo	VARCHAR(50)				Tipo de depósito

CATEGORIA	Tipo de dato	PK	FK	Referencia	Descripción
categoria_id	INT	✓			Identificador de categoría
tipo_producto	VARCHAR(100)				Tipo de producto

% GANANCIA	Tipo de dato	PK	FK	Referencia	Descripción
porcentaje_id	INT	✓			Identificador del porcentaje
porcentaje	DECIMAL(5,2)				Porcentaje de ganancia
categoria_id	INT		✓	Categoría(categoría_id)	Categoría asociada

PRODUCTO	Tipo de dato	PK	FK	Referencia	Descripción
producto_id	INT	✓			Identificador del producto
nombre	VARCHAR(100)				Nombre del producto
stock_minimo	INT				Stock mínimo
categoria_id	INT		✓	Categoría(categoría_id)	Categoría
reponer	BOOLEAN				Indica reposición

LISTA PRECIO	Tipo de dato	PK	FK	Referencia	Descripción
lista_id	INT	✓			Identificador de lista
producto_id	INT		✓	Producto(producto_id)	Producto
costo_unitario	DECIMAL(10,2)				Costo unitario
porcentaje_id	INT		✓	Porcentaje_gcia(porcentaje_id)	Porcentaje aplicado
precio_unitario	DECIMAL(10,2)				Precio final

VENDEDOR	Tipo de dato	PK	FK	Referencia	Descripción
vendedor_id	INT	✓			Identificador del vendedor
nombre	VARCHAR(100)				Nombre
apellido	VARCHAR(100)				Apellido
ranking	INT				Ranking
comision	DECIMAL(5,2)				Comisión
zona_id	INT		✓	Zona_venta(zona_id)	Zona asignada

ENVIO	Tipo de dato	PK	FK	Referencia	Descripción
envio_id	INT	✓			Identificador del envío
costo_envio	DECIMAL(10,2)				Costo
fecha_envio	DATE				Fecha de envío

VENTAS	Tipo de dato	PK	FK	Referencia	Descripción
venta_id	INT	✓			Identificador de venta
cliente_id	INT		✓	Cliente(cliente_id)	Cliente
fecha_venta	DATE				Fecha
metodo_pago_id	INT		✓	Metodo_pago(metodo_pago_id)	Método de pago
total	DECIMAL(10,2)				Total
estado_pago	VARCHAR(50)				Estado
vendedor_id	INT		✓	Vendedor(vendedor_id)	Vendedor
envio_id	INT		✓	Envio(envio_id)	Envío

DETALLE VTA	Tipo de dato	PK	FK	Referencia	Descripción
detallev_id	INT	✓			Identificador
venta_id	INT		✓	Venta(venta_id)	Venta
producto_id	INT		✓	Producto(producto_id)	Producto
cantidad	INT				Cantidad
precio_unitario	DECIMAL(10,2)				Precio
subtotal	DECIMAL(10,2)				Subtotal

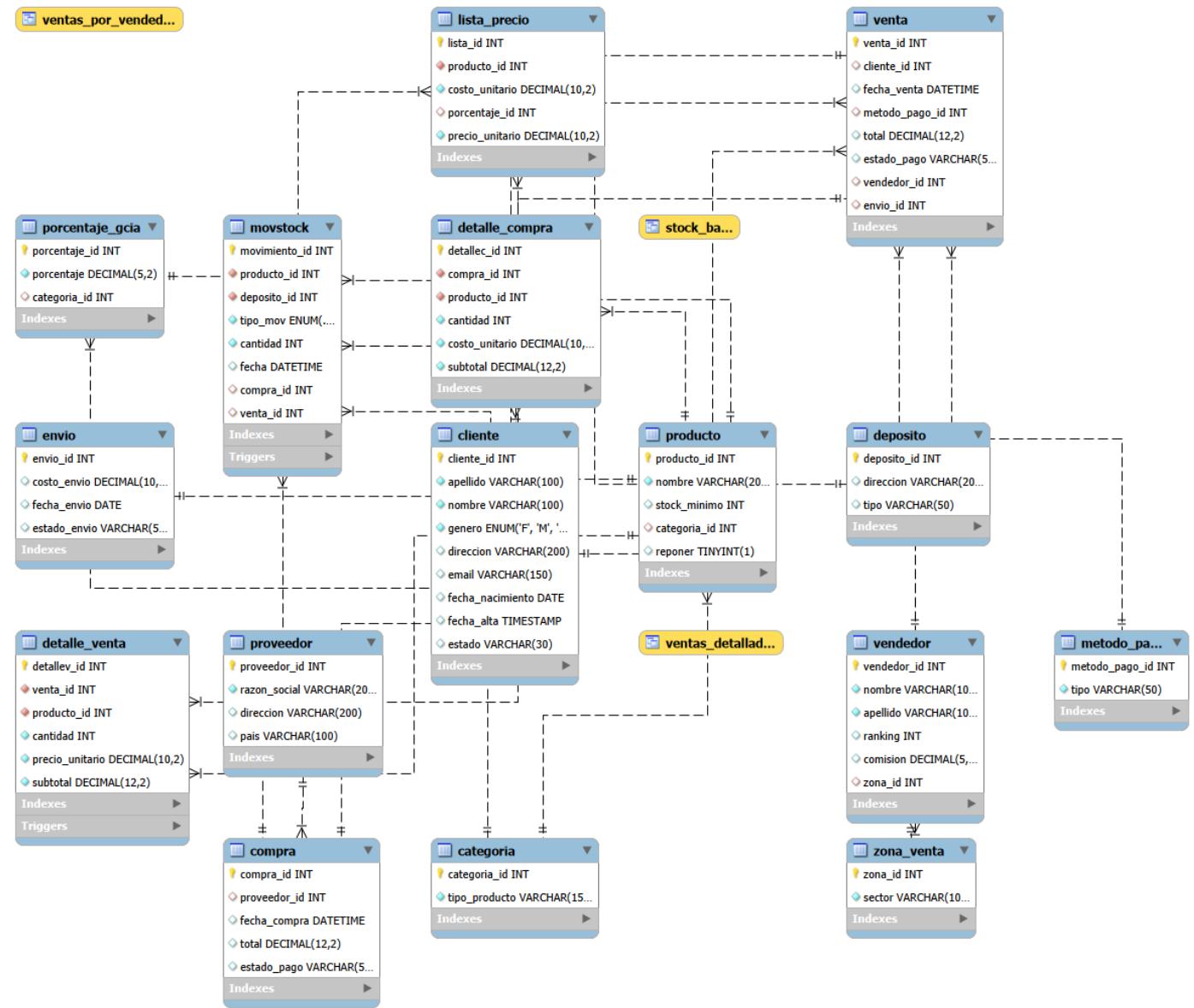
COMPRA	Tipo de dato	PK	FK	Referencia	Descripción
compra_id	INT	✓			Identificador
proveedor_id	INT		✓	Proveedor(proveedor_id)	Proveedor
fecha_compra	DATE				Fecha
total	DECIMAL(10,2)				Total
estado_pago	VARCHAR(50)				Estado

DETALLE CPRA	Tipo de dato	PK	FK	Referencia	Descripción
detallec_id	INT	✓			Identificador
compra_id	INT		✓	Compra(compra_id)	Compra
producto_id	INT		✓	Producto(producto_id)	Producto
cantidad	INT				Cantidad
costo_unitario	DECIMAL(10,2)				Costo
subtotal	DECIMAL(10,2)				Subtotal

Campo	Tipo de dato	PK	FK	Referencia	Descripción
movimiento_id	INT	✓			Movimiento
producto_id	INT		✓	Producto(producto_id)	Producto

<b>deposito_id</b>	INT		✓	Deposito(deposito_id)	Depósito
<b>tipo_mov</b>	VARCHAR(50)				Tipo de movimiento
<b>cantidad</b>	INT				Cantidad
<b>fecha</b>	DATE				Fecha
<b>compra_id</b>	INT		✓	Compra(compra_id)	Compra
<b>venta_id</b>	INT		✓	Venta(venta_id)	Venta

## DIAGRAMA ENTIDAD – RELACION



## VISTAS:

En una base de datos relacional, las vistas son consultas almacenadas que permiten presentar información específica de manera organizada y simplificada, sin modificar ni duplicar los datos originales. Funcionan como tablas virtuales que se generan a partir de una o más tablas reales. En el presente proyecto, las vistas fueron implementadas con el objetivo de optimizar el análisis comercial y el control de stock, permitiendo obtener información estratégica de manera rápida y eficiente.

A continuación, se describen las vistas implementadas en el Sistema:

### **Vw\_ventas\_detalladas**

La vista **ventas detalladas** permite visualizar el detalle completo de cada venta realizada por la empresa. Integra información proveniente de múltiples tablas, como ventas, clientes, productos y detalle\_venta, mostrando en una sola consulta datos como número de venta, fecha, cliente, producto vendido, cantidad, precio unitario y subtotal.

Su objetivo es facilitar el análisis operativo diario, permitiendo identificar qué productos se vendieron, en qué cantidades y a qué clientes. Esta vista resulta fundamental para el control comercial y la trazabilidad de cada operación.

### **Vw\_ventas\_por\_vendedor**

La vista **ventas\_por\_vendedor** consolida la información de ventas agrupada por vendedor. Su finalidad es medir el desempeño comercial del equipo de ventas.

Generalmente muestra el nombre del vendedor y el total vendido (ya sea en cantidad de operaciones o en monto total facturado). Es una herramienta clave para la toma de decisiones comerciales y la definición de incentivos o estrategias de mejora.

### **Vw\_stock\_bajo**

La vista **stock\_bajo** permite detectar productos cuyo nivel de stock se encuentra por debajo del mínimo establecido.

Su objetivo es apoyar la gestión operativa del inventario, alertando sobre productos que requieren reposición urgente. Con esta información, la empresa puede anticiparse a problemas de abastecimiento y mantener un control eficiente del inventario.

## STORED PROCEDURES:

Los stored procedures permiten automatizar operaciones frecuentes, centralizar la lógica del sistema y mejorar la seguridad, ya que evitan que los usuarios ejecuten directamente múltiples instrucciones sobre las tablas.

En este proyecto se implementaron los siguientes procedimientos:

#### **Sp\_registrar\_venta**

Este procedimiento se encarga de registrar una venta, insertando los datos correspondientes en la tabla venta y su detalle asociado. Su objetivo es centralizar el proceso de alta de ventas y evitar inconsistencias.

Tablas utilizadas: venta y detalle\_venta

#### **Sp\_actualizar\_stock**

Este procedimiento registra movimientos de stock automáticamente a partir de una compra o una venta. Su objetivo es mantener actualizado el inventario del sistema.

Tablas utilizadas: movstock y product

### **FUNCIONES:**

Las funciones en SQL son objetos almacenados dentro de la base de datos que permiten realizar cálculos o procesos específicos y devolver un único valor como resultado. A diferencia de los procedimientos almacenados, las funciones siempre retornan un valor y pueden utilizarse directamente dentro de consultas

#### **fn\_calcular\_subtotal\_venta**

Esta función calcula el subtotal de un producto vendido multiplicando la cantidad por el precio unitario. Su objetivo es estandarizar el cálculo del subtotal en los registros de detalle de venta.

Tablas utilizadas: detalle\_venta (cantidad y precio\_unitario)

#### **fn\_total\_vendido\_producto**

Devuelve la cantidad total de unidades vendidas de un producto específico. Recibe como parámetro el identificador del producto y calcula la suma de las cantidades registradas en la tabla detalle\_venta.

Se utiliza la función SUM junto con COALESCE para evitar valores nulos en caso de que el producto no registre ventas.

Esta función permite obtener información rápida sobre el desempeño comercial de un producto sin necesidad de realizar consultas complejas.

## TRIGGER:

Es un objeto de la base de datos que se ejecuta automáticamente cuando ocurre un evento específico sobre una tabla, como una inserción, actualización o eliminación.

Su principal característica es que no se ejecuta manualmente, sino que se activa de forma automática ante una acción determinada.

### **trg\_descuento\_stock**

Se ejecuta automáticamente después de insertar un registro en la tabla detalle de venta. Su función es registrar de manera automática la salida de stock correspondiente a cada producto vendido. Este mecanismo permite que cada venta genere automáticamente su impacto en el inventario, evitando que el descuento de stock dependa de una acción manual. De esta manera, se garantiza la coherencia entre ventas y movimientos de stock.

### **trg\_control\_stock**

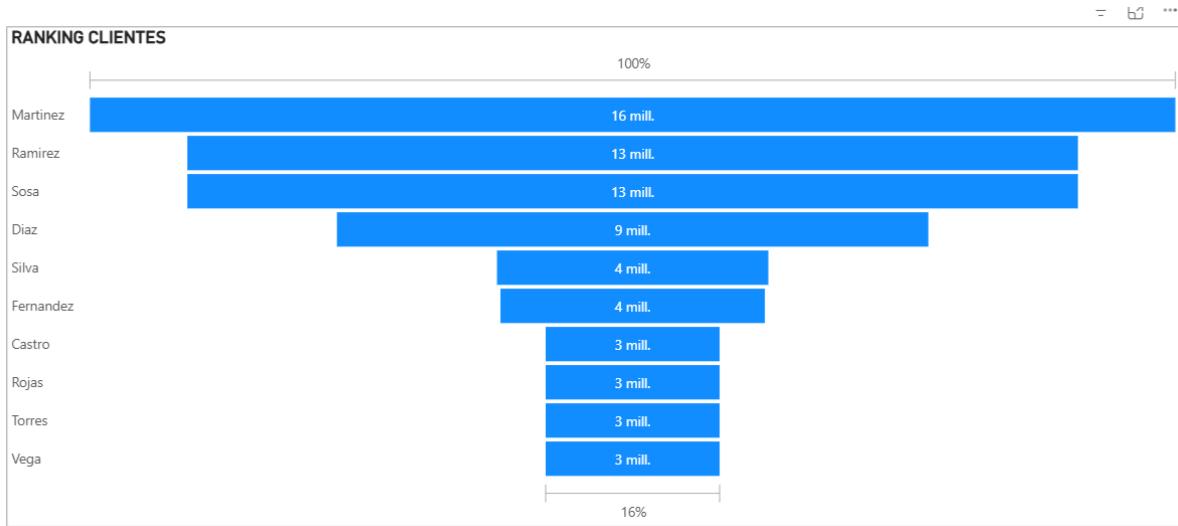
Se ejecuta automáticamente después de insertar un movimiento en la table movstock. Su objetivo es controlar el nivel de stock y marcar aquellos productos que necesitan reposición. Este trigger cumple una función preventiva, ya que permite identificar automáticamente productos con stock crítico, facilitando la planificación de compras y evitando faltantes.

## INTELIGENCIA APICADA A LOS NEGOCIOS (BI)

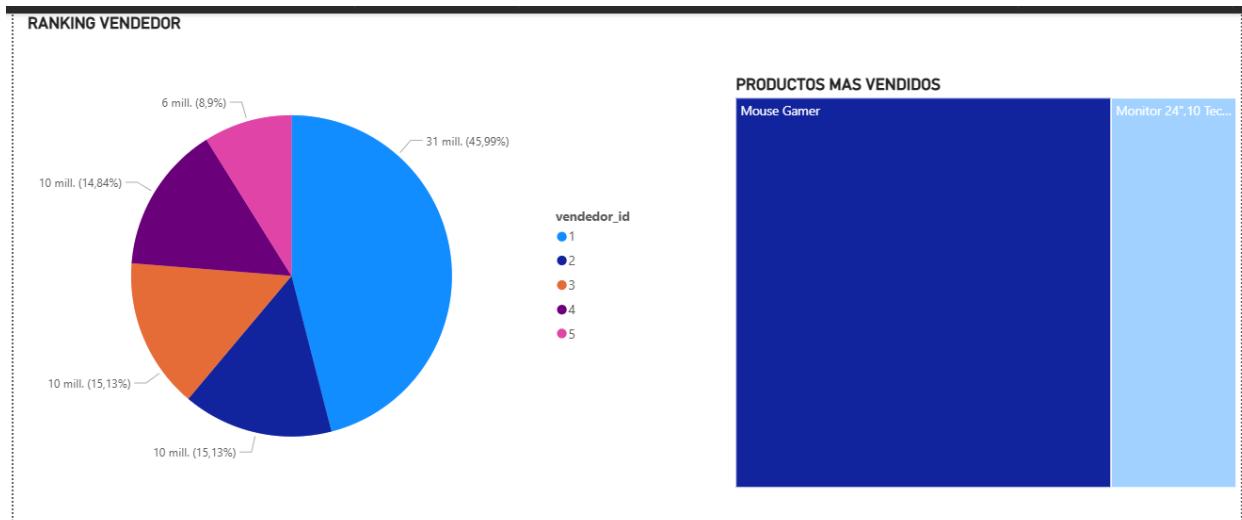
Con el objetivo de transformar los datos operativos del sistema de ventas en información estratégica, se realizó un análisis mediante Power BI utilizando los datos almacenados en la base de datos desarrollada.

A través de la conexión con las tablas y vistas del sistema, se construyeron visualizaciones que permiten interpretar el comportamiento comercial de la empresa, evaluar el rendimiento de vendedores y clientes, analizar la rotación de productos y controlar el estado del stock.

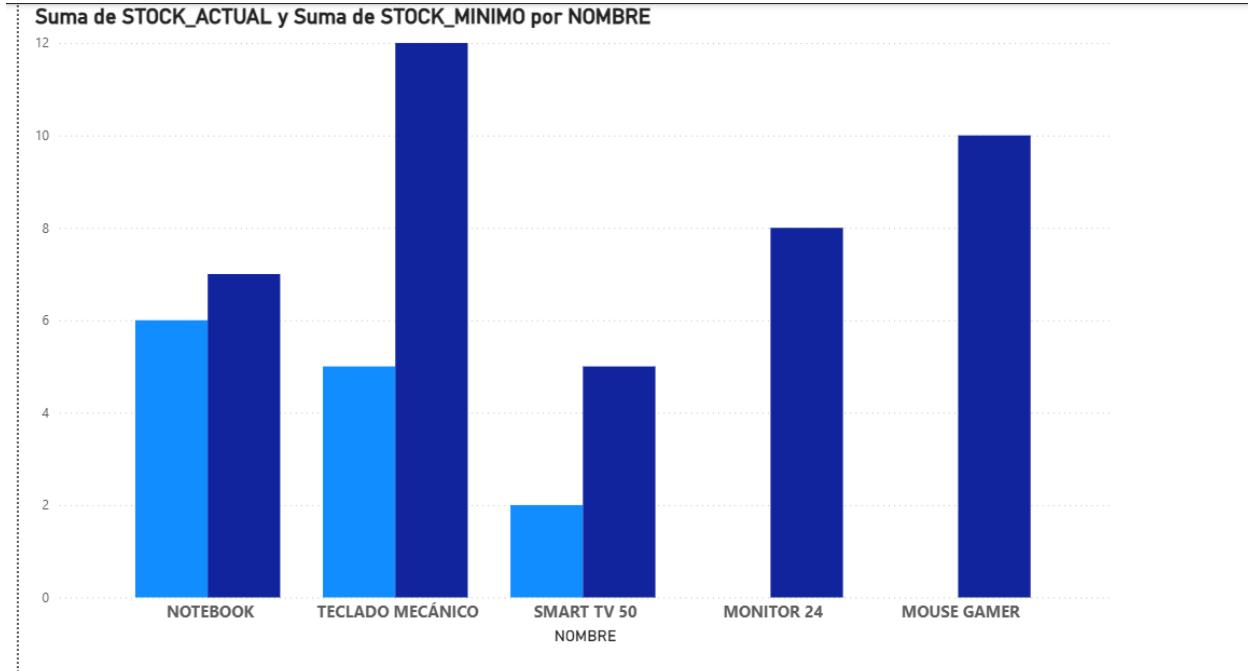
**Análisis ranking de clientes:** El análisis de ranking de clientes permitió identificar aquellos que generan mayor facturación para la empresa. A través del cálculo del total gastado y la cantidad de compras realizadas, se determinó cuáles son los clientes más valiosos desde el punto de vista comercial. Con estos indicadores podremos diseñar estrategias de fidelización y ofrecer beneficios.



**Ranking de vendedores:** El análisis de desempeño de vendedores permitió evaluar el rendimiento comercial individual en función del total vendido. Nos permite medir productividad, comparar el desempeño y establecer incentivos y comisiones basadas en resultados. El análisis de desempeño de vendedores permitió evaluar el rendimiento comercial individual en función del total vendido.



**Stock mínimo:** El análisis de stock mínimo permitió identificar productos cuyo nivel de inventario se encuentra en estado crítico o cercano al mínimo establecido. Con esto se logra anticiparse a necesidades de reposición, reducir el riesgo de faltantes y mejorar la planificación de compras. Este análisis es clave para la gestión operativa, ya que un control eficiente del inventario impacta directamente en la continuidad de ventas y en la satisfacción del cliente.



Antes de la entrega final, se realizó la eliminación y recreación completa del esquema de base de datos para verificar que el script se ejecute correctamente desde cero, sin errores ni dependencias faltantes. Se realizó una copia de seguridad completa del esquema y los datos mediante exportación del script SQL, garantizando la posibilidad de restauración ante cualquier inconveniente.

Esta práctica asegura la integridad y recuperación del sistema.

El desarrollo del sistema permitió integrar modelado de datos, automatización mediante triggers y funciones, y análisis estratégico a través de herramientas de Business Intelligence.

La solución implementada demuestra cómo una base de datos correctamente diseñada puede convertirse en una herramienta clave para la gestión comercial y el control operativo de una empresa.