

UNIVERSIDAD PRIVADA “FRANZ TAMAYO”



CARRERA: INGENIERIA EN SISTEMAS

ESTUDIANTES: ELIO VASQUEZ GONZALES
LUIS FERNANDO MAMANI CHQUE
SANTOS BRAYAN HUMIRI QUISPE
ADRIAN RAFAEL MARTINEZ QUIROGA

DOCENTE: ING. WILLIAM RODDY BARRA PAREDES

ACTIVIDAD: “INFORME DE PROYECTO - HITO5”

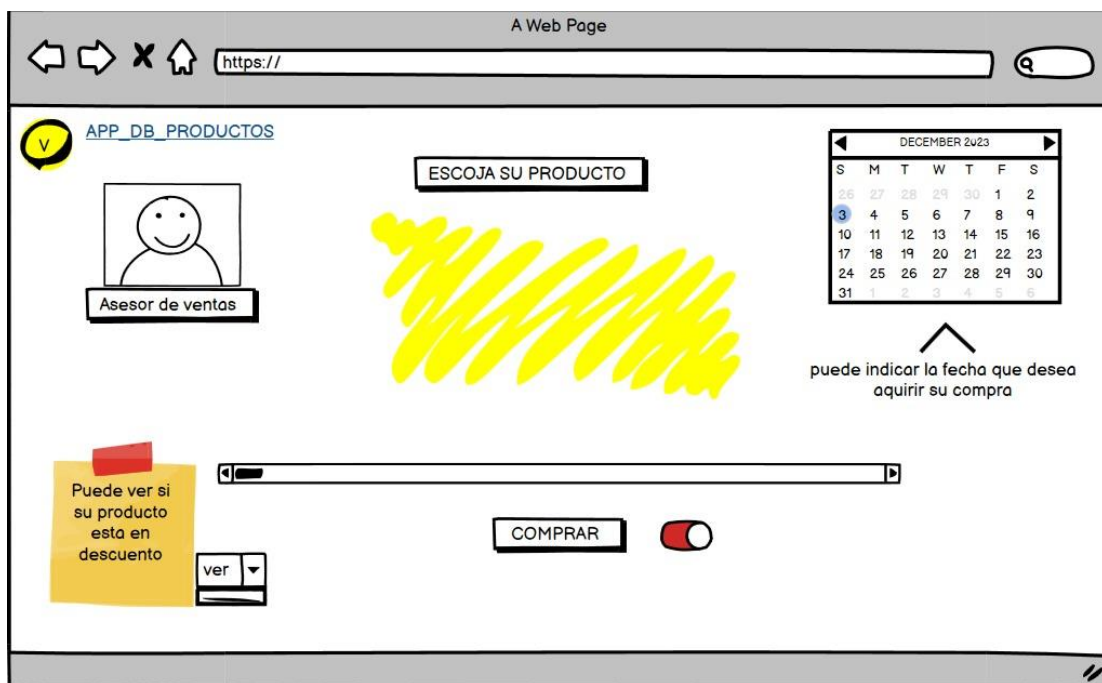
1. INTRODUCCIÓN

El propósito de este proyecto es desarrollar un sistema eficiente de gestión de productos, enfocado en mejorar la experiencia de los usuarios al buscar y adquirir productos específicos. La idea es permitir a cualquier persona encontrar y comprar productos que se ajusten a sus necesidades y preferencias, tomando en cuenta su ubicación actual. De esta manera, el sistema proporcionará recomendaciones personalizadas y opciones de compra en establecimientos cercanos o con opciones de entrega rápida, asegurando una experiencia de compra más conveniente.

2. ANÁLISIS Y DISEÑO DE LA BASE DE DATOS

a. Contexto de la Base de Datos:

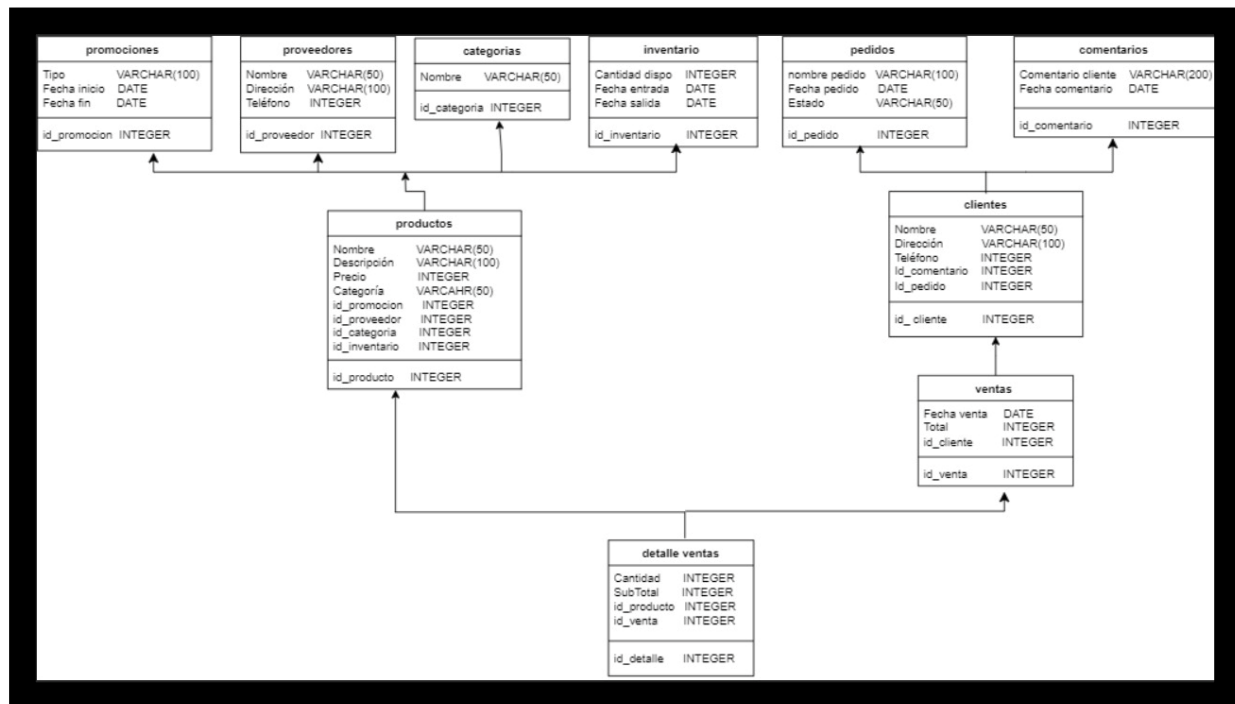
Dado el enfoque en la gestión eficiente de productos, y considerando la necesidad de localizar y administrar una variedad de productos en relación con las preferencias y ubicaciones de los usuarios, hemos determinado que el nombre más adecuado para nuestra base de datos será **APP_DB_PRODUCTOS**. Este nombre refleja claramente el propósito de la base de datos, que es centralizar y organizar la información sobre productos disponibles, facilitando así un sistema de gestión de productos más integrado y accesible para los usuarios.



b. Entidades/tablas de sistema: Las tablas necesarias que se identificaron para nuestro sistema son las siguientes:

TABLA	FUNCION DE LA TABLA
PROMOCIONES	Almacena información sobre promociones, como el tipo de promoción, fechas de inicio y fin, e identificación única.
PROVEEDORES	Contiene detalles sobre los proveedores, incluyendo nombre, dirección, número de teléfono y una identificación única.
CATEGORÍAS	Guarda las categorías de productos disponibles en el sistema con un nombre y una identificación única.
INVENTARIO	Registra detalles del inventario, como la cantidad disponible, fechas de entrada y salida, y una identificación única.
PRODUCTOS	Almacena información detallada sobre los productos, incluyendo nombre, descripción, precio, categoría, y establece relaciones con otras tablas mediante claves foráneas.
PEDIDOS	Mantiene detalles de los pedidos realizados, como el nombre del pedido, fecha, estado y una identificación única.
COMENTARIOS	Almacena comentarios de clientes junto con la fecha y una identificación única.
CLIENTES	Contiene información sobre los clientes, como nombre, dirección, teléfono, y establece relaciones con pedidos y comentarios mediante claves foráneas.
VENTAS	Registra las ventas realizadas, incluyendo la fecha, total y se relaciona con la tabla clientes mediante una clave foránea.
DETALLE_VENTAS	Guarda detalles específicos de las ventas, como la cantidad, subtotal, y se relaciona con productos y ventas mediante claves foráneas.

c. Diseño de la base de datos.



https://drive.google.com/file/d/1hGZ6umqe60320FQ73m3UvFndE_TxBU5T/view

CÓDIGO SQL.

BASE DE DATOS:

```
CREATE DATABASE APP_DB_PRODUCTOS
USE APP_DB_PRODUCTOS
```

TABLAS Y REGISTROS:

```
CREATE TABLE promociones
(
    tipo VARCHAR(100) NOT NULL,
    fecha_inicio DATE NOT NULL,
    fecha_fin DATE NOT NULL,
    id_promocion INTEGER PRIMARY KEY NOT NULL
);
```

```
INSERT INTO promociones (tipo, fecha_inicio, fecha_fin, id_promocion)
VALUES ('Descuento', '2023-12-01', '2023-12-15', 1);
INSERT INTO promociones (tipo, fecha_inicio, fecha_fin, id_promocion)
VALUES ('Oferta Especial', '2023-12-05', '2023-12-20', 2);
INSERT INTO promociones (tipo, fecha_inicio, fecha_fin, id_promocion)
VALUES ('Rebaja de Temporada', '2023-12-10', '2023-12-31', 3);
INSERT INTO promociones (tipo, fecha_inicio, fecha_fin, id_promocion)
VALUES ('Cupón de Regalo', '2023-12-15', '2023-12-25', 4);
```

```
CREATE TABLE proveedores
(
    nombre VARCHAR(100) NOT NULL,
    direccion VARCHAR(200) NOT NULL,
```

```
telefono INTEGER NOT NULL,  
id_proveedor INTEGER PRIMARY KEY NOT NULL  
);
```

```
INSERT INTO proveedores (nombre, direccion, telefono, id_proveedor)  
VALUES ('Proveedor A', '123 Calle Principal', 123456, 1);  
INSERT INTO proveedores (nombre, direccion, telefono, id_proveedor)  
VALUES ('Proveedor B', '456 Avenida Secundaria', 987654, 2);  
INSERT INTO proveedores (nombre, direccion, telefono, id_proveedor)  
VALUES ('Proveedor C', '789 Carretera Central', 246810, 3);  
INSERT INTO proveedores (nombre, direccion, telefono, id_proveedor)  
VALUES ('Proveedor D', '012 Calle Secundaria', 3691234, 4);
```

CREATE TABLE categorias

```
(  
nombre VARCHAR(50) NOT NULL,  
id_categoria INTEGER PRIMARY KEY NOT NULL  
);
```

```
INSERT INTO categorias (nombre, id_categoria)  
VALUES ('Electrónicos', 1);  
INSERT INTO categorias (nombre, id_categoria)  
VALUES ('Ropa', 2);  
INSERT INTO categorias (nombre, id_categoria)  
VALUES ('Hogar', 3);  
INSERT INTO categorias (nombre, id_categoria)  
VALUES ('Alimentos', 4);
```

CREATE TABLE inventario

```
(  
cantidad_disp INTEGER NOT NULL,
```

```
fecha_entrada DATE NOT NULL,  
fecha_salida DATE NOT NULL,  
id_inventario INTEGER PRIMARY KEY NOT NULL  
);
```

```
INSERT INTO inventario (cantidad_disp, fecha_entrada, fecha_salida,  
id_inventario)  
VALUES (100, '2023-11-01', '2023-12-01', 1);  
INSERT INTO inventario (cantidad_disp, fecha_entrada, fecha_salida,  
id_inventario)  
VALUES (150, '2023-11-15', '2023-12-15', 2);  
INSERT INTO inventario (cantidad_disp, fecha_entrada, fecha_salida,  
id_inventario)  
VALUES (200, '2023-11-30', '2023-12-30', 3);  
INSERT INTO inventario (cantidad_disp, fecha_entrada, fecha_salida,  
id_inventario)  
VALUES (120, '2023-12-05', '2023-12-31', 4);
```

```
CREATE TABLE productos  
(  
    nombre VARCHAR(100) NOT NULL,  
    descripcion VARCHAR(100) NOT NULL,  
    precio INTEGER NOT NULL,
```



```

categoria VARCHAR(50) NOT NULL,
id_promocion INTEGER NOT NULL,
id_proveedor INTEGER NOT NULL,
id_categoria INTEGER NOT NULL,
id_inventario INTEGER NOT NULL,
id_producto INT PRIMARY KEY NOT NULL,
FOREIGN KEY (id_promocion) REFERENCES promociones(id_promocion),
FOREIGN KEY (id_proveedor) REFERENCES proveedores(id_proveedor),
FOREIGN KEY (id_categoria) REFERENCES categorias(id_categoria),
FOREIGN KEY (id_inventario) REFERENCES inventario(id_inventario)
);

```

```

INSERT INTO productos (nombre, descripcion, precio, categoria, id_promocion,
id_proveedor, id_categoria, id_inventario, id_producto)
VALUES ('Smartphone', 'Último modelo con cámara de alta resolución', 800,
'Electrónicos', 1, 1, 1, 1, 1);
INSERT INTO productos (nombre, descripcion, precio, categoria, id_promocion,
id_proveedor, id_categoria, id_inventario, id_producto)
VALUES ('Camisa', 'Camisa de algodón de alta calidad', 50, 'Ropa', 2, 2, 2, 2, 2);
INSERT INTO productos (nombre, descripcion, precio, categoria, id_promocion,
id_proveedor, id_categoria, id_inventario, id_producto)
VALUES ('Sofá', 'Sofá de tres plazas con diseño moderno', 700, 'Hogar', 3, 3, 3, 3,
3);
INSERT INTO productos (nombre, descripcion, precio, categoria, id_promocion,
id_proveedor, id_categoria, id_inventario, id_producto)
VALUES ('Arroz', 'Arroz de grano largo, paquete de 5kg', 10, 'Alimentos', 4, 4, 4, 4,
4);

```

CREATE TABLE pedidos

```

(
nombre_pedido VARCHAR(100) NOT NULL,
fecha_pedido DATE NOT NULL,
estado VARCHAR(50) NOT NULL,

```

```
id_pedido INTEGER PRIMARY KEY NOT NULL
);
```

```
INSERT INTO pedidos (nombre_pedido, fecha_pedido, estado, id_pedido)
VALUES ('Pedido 1', '2023-12-01', 'En Proceso', 1);
INSERT INTO pedidos (nombre_pedido, fecha_pedido, estado, id_pedido)
VALUES ('Pedido 2', '2023-12-05', 'En Proceso', 2);
INSERT INTO pedidos (nombre_pedido, fecha_pedido, estado, id_pedido)
VALUES ('Pedido 3', '2023-12-10', 'Entregado', 3);
INSERT INTO pedidos (nombre_pedido, fecha_pedido, estado, id_pedido)
VALUES ('Pedido 4', '2023-12-15', 'En Proceso', 4);
```

CREATE TABLE comentarios

```
(
    comentario_cliente VARCHAR(200) NOT NULL,
    fecha_comentario DATE NOT NULL,
    id_comentario INTEGER PRIMARY KEY NOT NULL
);
```

```
INSERT INTO comentarios (comentario_cliente, fecha_comentario, id_comentario)
VALUES ('Excelente servicio, muy satisfecho con mi compra.', '2023-12-01', 1);
INSERT INTO comentarios (comentario_cliente, fecha_comentario, id_comentario)
VALUES ('El producto llegó en mal estado, espero una pronta solución.', '2023-12-05', 2);
INSERT INTO comentarios (comentario_cliente, fecha_comentario, id_comentario)
VALUES ('Buena atención al cliente, definitivamente volveré a comprar.', '2023-12-10', 3);
INSERT INTO comentarios (comentario_cliente, fecha_comentario, id_comentario)
VALUES ('El envío se retrasó, necesito una explicación.', '2023-12-15', 4);
```

CREATE TABLE clientes

```
(
    nombre VARCHAR(50) NOT NULL,
    direccion VARCHAR(100) NOT NULL,
```

```
telefono INTEGER NOT NULL,  
id_pedido INTEGER NOT NULL,  
id_comentario INTEGER NOT NULL,  
id_cliente INTEGER PRIMARY KEY NOT NULL,  
FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido),  
FOREIGN KEY (id_comentario) REFERENCES comentarios(id_comentario)  
);
```

```
INSERT INTO clientes (nombre, direccion, telefono, id_pedido, id_comentario,  
id_cliente)  
VALUES ('Cliente A', '456 Calle Principal', 123456, 1, 1, 11);  
INSERT INTO clientes (nombre, direccion, telefono, id_pedido, id_comentario,  
id_cliente)  
VALUES ('Cliente B', '789 Avenida Secundaria', 98765, 2, 2, 22);  
INSERT INTO clientes (nombre, direccion, telefono, id_pedido, id_comentario,  
id_cliente)  
VALUES ('Cliente C', '012 Carretera Central', 246810, 3, 3, 33);  
INSERT INTO clientes (nombre, direccion, telefono, id_pedido, id_comentario,  
id_cliente)  
VALUES ('Cliente D', '123 Calle Secundaria', 36912, 4, 4, 44);
```

CREATE TABLE ventas

```
(  
    fecha_venta DATE NOT NULL,  
    total INTEGER NOT NULL,  
    id_cliente INTEGER NOT NULL,  
    id_venta INTEGER PRIMARY KEY NOT NULL,  
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)  
);
```

```
INSERT INTO ventas (fecha_venta, total, id_cliente, id_venta)  
VALUES ('2023-12-01', 800, 11, 12);  
INSERT INTO ventas (fecha_venta, total, id_cliente, id_venta)  
VALUES ('2023-12-05', 50, 22, 23);  
INSERT INTO ventas (fecha_venta, total, id_cliente, id_venta)  
VALUES ('2023-12-10', 700, 33, 34);  
INSERT INTO ventas (fecha_venta, total, id_cliente, id_venta)  
VALUES ('2023-12-15', 10, 44, 45);
```

CREATE TABLE detalle_ventas

```
(  
    cantidad INTEGER NOT NULL,  
    subtotal INTEGER NOT NULL,  
    id_producto INTEGER NOT NULL,
```

```
id_venta INTEGER NOT NULL,  
id_detalle INTEGER PRIMARY KEY NOT NULL,  
FOREIGN KEY (id_producto) REFERENCES productos(id_producto),  
FOREIGN KEY (id_venta) REFERENCES ventas(id_venta)  
);
```

```
INSERT INTO detalle_ventas (cantidad, subtotal, id_producto, id_venta, id_detalle)  
VALUES (2, 1600, 1, 12, 111);  
INSERT INTO detalle_ventas (cantidad, subtotal, id_producto, id_venta, id_detalle)  
VALUES (1, 50, 2, 23, 222);  
INSERT INTO detalle_ventas (cantidad, subtotal, id_producto, id_venta, id_detalle)  
VALUES (1, 700, 3, 34, 333);  
INSERT INTO detalle_ventas (cantidad, subtotal, id_producto, id_venta, id_detalle)  
VALUES (1, 10, 4, 45, 444);
```

“CONSULTAS SQL”

```
SELECT * FROM promociones;  
SELECT * FROM proveedores;
```

```
SELECT * FROM categorias;  
SELECT * FROM inventario;  
SELECT * FROM productos;  
SELECT * FROM pedidos;  
SELECT * FROM comentarios;  
SELECT * FROM clientes;  
SELECT * FROM ventas;  
SELECT * FROM detalle_ventas;
```

---4 JOINS CON MÁS DE 5 TABLAS:---

--1.¿Cuáles fueron las ventas realizadas el 5 de diciembre de 2023, incluyendo información detallada sobre los clientes

--, comentarios, productos y cantidades vendidas?

```
SELECT  venta.fecha_venta,  venta.total,  cliente.nombre AS cliente,  
comentario.comentario_cliente, producto.nombre AS producto, detalle.cantidad  
FROM ventas AS venta  
JOIN clientes AS cliente ON venta.id_cliente = cliente.id_cliente  
JOIN  comentarios AS comentario ON cliente.id_comentario =  
comentario.id_comentario  
JOIN detalle_ventas AS detalle ON venta.id_venta = detalle.id_venta  
JOIN productos AS producto ON detalle.id_producto = producto.id_producto  
WHERE venta.fecha_venta = '2023-12-05';
```

---2.Buscar información de una venta específica(con id = 34), incluyendo clientes, comentarios, productos y detalles de ventas:

```
SELECT  venta.fecha_venta,  venta.total,  cliente.nombre AS cliente,  
comentario.comentario_cliente, producto.nombre AS producto, detalle.cantidad  
FROM ventas AS venta  
JOIN clientes AS cliente ON venta.id_cliente = cliente.id_cliente  
JOIN  comentarios AS comentario ON cliente.id_comentario =  
comentario.id_comentario  
JOIN detalle_ventas AS detalle ON venta.id_venta = detalle.id_venta  
JOIN productos AS producto ON detalle.id_producto = producto.id_producto  
WHERE venta.id_venta = 34;
```

---3.Buscar detalles de inventario, incluyendo productos, proveedores, promociones y categorías:

```

SELECT    producto.nombre    AS    producto,    inventario.cantidad_disp,
proveedor.nombre    AS    proveedor,    promocion.tipo    AS    tipo_promocion,
categoria.nombre AS categoria
FROM productos AS producto
JOIN inventario ON producto.id_inventario = inventario.id_inventario
JOIN    proveedores    AS    proveedor    ON    producto.id_proveedor    =
proveedor.id_proveedor
JOIN    promociones    AS    promocion    ON    producto.id_promocion    =
promocion.id_promocion
JOIN categorias AS categoria ON producto.id_categoria = categoria.id_categoria
WHERE producto.id_producto = 4;

```

**---4. "¿Cuáles son los productos en inventario con una cantidad disponible superior a 50 unidades, incluyendo
--detalles sobre el nombre del producto, descripción, proveedor, categoría y la cantidad disponible?"**

```

SELECT
    inventario.fecha_entrada,
    producto.nombre AS producto,
    producto.descripcion,
    proveedor.nombre AS proveedor,
    categorias.nombre AS categoria,
    inventario.cantidad_disp,
    promocion.tipo AS tipo_promocion,
    promocion.fecha_inicio,
    promocion.fecha_fin
FROM productos AS producto
JOIN    proveedores    AS    proveedor    ON    producto.id_proveedor    =
proveedor.id_proveedor
JOIN categorias ON producto.id_categoria = categorias.id_categoria
JOIN inventario ON producto.id_inventario = inventario.id_inventario
JOIN    promociones    AS    promocion    ON    producto.id_promocion    =
promocion.id_promocion
WHERE inventario.cantidad_disp > 50;

```

--3 JOINS CON MAS DE 2 TABLAS--

**--1.¿Podría proporcionar detalles sobre las ventas de productos electrónicos,
--incluyendo la fecha de venta, el nombre del producto y el nombre del cliente
asociado?**

```
SELECT
    v.fecha_venta,
    p.nombre AS producto,
    c.nombre AS cliente
FROM
    ventas v
INNER JOIN detalle_ventas d ON v.id_venta = d.id_venta
INNER JOIN productos p ON d.id_producto = p.id_producto
INNER JOIN clientes c ON v.id_cliente = c.id_cliente
INNER JOIN categorias cat ON p.id_categoria = cat.id_categoria
WHERE cat.nombre = 'Electrónicos';
```

**--2.¿Cuáles son los detalles de los pedidos realizados, incluyendo el
--nombre del cliente, el estado del pedido y cualquier comentario asociado?**

```
SELECT
    v.fecha_venta,
    v.total,
    d.cantidad,
    p.nombre AS producto,
    c.nombre AS cliente
FROM
    ventas v
INNER JOIN detalle_ventas d ON v.id_venta = d.id_venta
INNER JOIN productos p ON d.id_producto = p.id_producto
INNER JOIN clientes c ON v.id_cliente = c.id_cliente;
```

**--- 3.¿Podría proporcionar información detallada sobre las ventas realizadas,
--incluyendo la fecha de venta, el total, la cantidad y el nombre del producto
vendido, así como el nombre del cliente asociado?**

```
SELECT
    p.nombre AS producto,
    p.descripcion,
```

```
pr.nombre AS proveedor,  
c.nombre AS categoria,  
pm.tipo AS tipo_promocion  
FROM  
    productos p  
INNER JOIN promociones pm ON p.id_promocion = pm.id_promocion  
INNER JOIN proveedores pr ON p.id_proveedor = pr.id_proveedor  
INNER JOIN categorias c ON p.id_categoria = c.id_categoria;
```

-----FUNCIONES DE AGREGACIÓN = 4 EJEMPLOS SIMPLES-----

---1.¿Cuántos productos están actualmente en oferta?

```
SELECT COUNT(*) as productos_en_oferta
```



```
FROM productos
WHERE id_promocion IS NOT NULL;
```

---2.¿Cuántos pedidos se encuentran actualmente en proceso?

```
SELECT COUNT(*) as pedidos_en_proceso
FROM pedidos
WHERE estado = 'En Proceso';
```

---3.¿Cuál fue la venta más grande en términos de monto total?

```
SELECT MAX(total) as venta_mas_grande
FROM ventas;
```

----4.¿Cuál es el promedio de ventas diarias?

```
SELECT AVG(total) as promedio_ventas_diarias
FROM ventas;
```

-FUNCIONES DE AGREGACIÓN II = 5 QUE MANEJE JOINS CON MAS DE 3 TABLAS-

--1.¿Cuántos productos diferentes se han vendido en cada categoría durante el último mes?

```
SELECT      categorias.nombre      AS      categoria,      COUNT(DISTINCT
productos.id_producto) AS productos_vendidos
FROM categorias
JOIN productos ON categorias.id_categoria = productos.id_categoria
JOIN detalle_ventas ON productos.id_producto = detalle_ventas.id_producto
JOIN ventas ON detalle_ventas.id_venta = ventas.id_venta
WHERE ventas.fecha_venta BETWEEN '2023-11-01' AND '2023-12-01'
GROUP BY categorias.nombre;
```

--2.¿Cuál es el total de compras y el promedio de compras por cliente?

```
SELECT clientes.nombre AS cliente, COUNT(DISTINCT ventas.id_venta) AS
total_compras, AVG(ventas.total) AS promedio_compras
FROM clientes
JOIN pedidos ON clientes.id_pedido = pedidos.id_pedido
JOIN ventas ON clientes.id_cliente = ventas.id_cliente
GROUP BY clientes.nombre;
```

--3. ¿Cuál es la cantidad total de productos en inventario suministrados por cada proveedor?

```
SELECT proveedores.nombre AS proveedor, SUM(inventario.cantidad_disp) AS
total_en_inventario
FROM proveedores
JOIN productos ON proveedores.id_proveedor = productos.id_proveedor
JOIN inventario ON productos.id_inventario = inventario.id_inventario
GROUP BY proveedores.nombre;
```

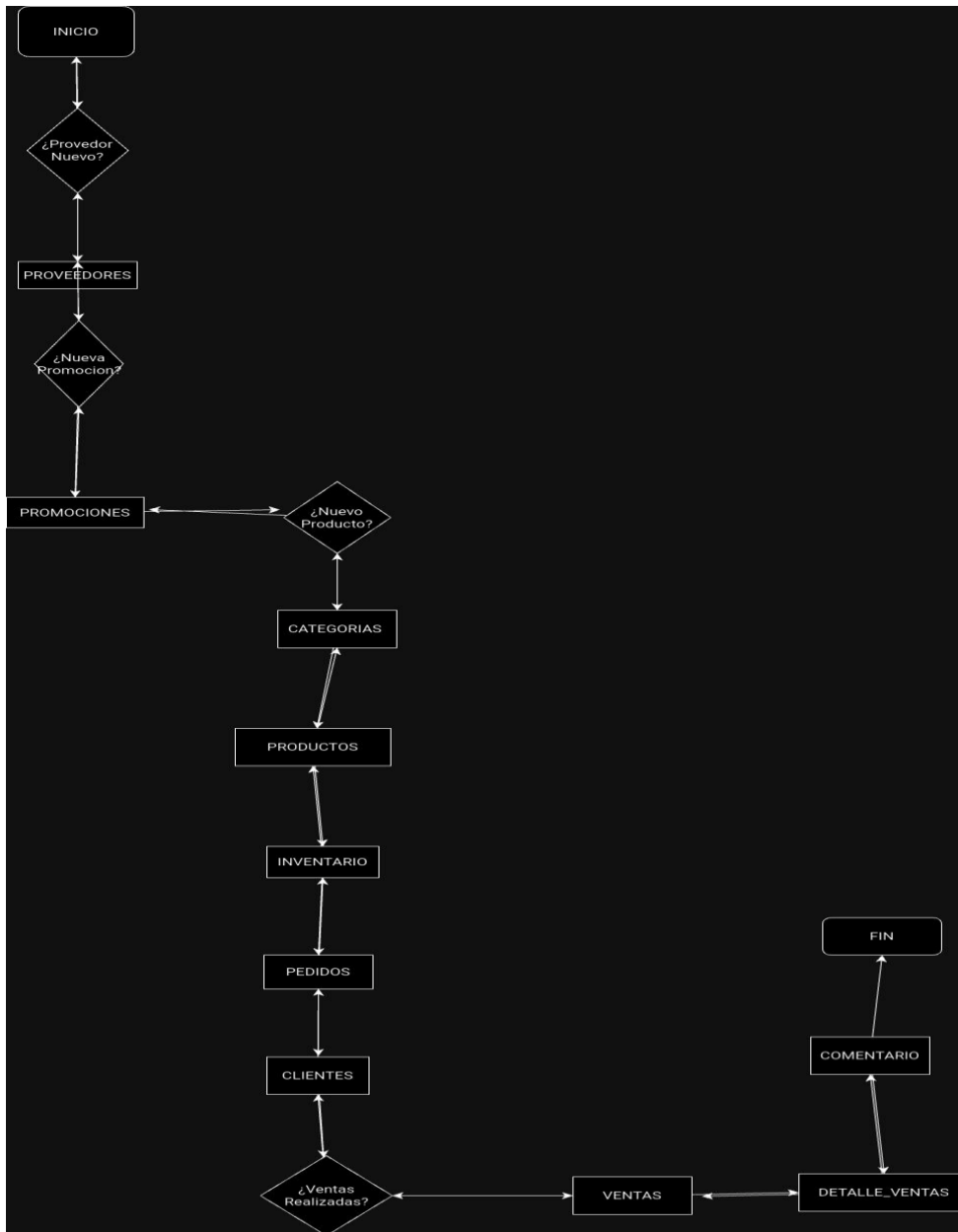
-- 4.¿Cuál es la cantidad total y el precio promedio de productos en oferta por categoría?

```
SELECT categorias.nombre AS categoria, COUNT(productos.id_producto) AS
total_ofertas, AVG(productos.precio) AS precio_promedio
FROM categorias
JOIN productos ON categorias.id_categoria = productos.id_categoria
WHERE productos.id_promocion IS NOT NULL
GROUP BY categorias.nombre;
```

---5.¿Cuánto ha comprado cada cliente de cada proveedor?

```
SELECT clientes.nombre AS cliente, proveedores.nombre AS proveedor,
SUM(ventas.total) AS total_comprado
FROM clientes
JOIN pedidos ON clientes.id_pedido = pedidos.id_pedido
JOIN ventas ON clientes.id_cliente = ventas.id_cliente
JOIN detalle_ventas ON ventas.id_venta = detalle_ventas.id_venta
JOIN productos ON detalle_ventas.id_producto = productos.id_producto
JOIN proveedores ON productos.id_proveedor = proveedores.id_proveedor
GROUP BY clientes.nombre, proveedores.nombre;
```

Modelo Lógico



I7ynZOnn6bLz7dUJA53dhWxh4qPtCQRusGhnJoaSsxq8EW5jts33Jt%2BVBQh7B
%2BxmR2li9V%2FaqxYqjVQAmhNoOuqVnSgVIYWQauitlMNR9%2BRfd58N9%2F
e82sOhDMt772%2Bg%2Bdq6v%2BP4ayGgSZZlhJawUJOlbqpDWDyghmmVNPWv
5dzZvu71UA5wXR2K42lekXRECzX7LKVuBte76qMNs9smTc8lYUFB9xaAza1oi0N
Xgxu9qiumV%2FfgQDrfX%2BQoJJvxx6gOjnc%2FHBWEIVtfn4M9%2F8D

3. USABILIDAD

126 %

Results										Messages									
</																			

--¿Cuáles fueron las ventas realizadas el 5 de diciembre de 2023, incluyendo información detallada sobre los clientes
--, comentarios, productos y cantidades vendidas?

----Buscar información de una venta específica(con id = 34), incluyendo clientes, comentarios, productos y detalles de ventas:
SELECT venta.fecha_venta, venta.total, cliente.nombre AS cliente, comentario.comentario_cliente, producto.nombre AS producto, detalle.cantidad
FROM ventas AS venta
JOIN clientes AS cliente ON venta.id_cliente = cliente.id_cliente
JOIN comentarios AS comentario ON cliente.id_comentario = comentario.id_comentario
JOIN detalle_ventas AS detalle ON venta.id_venta = detalle.id_venta
JOIN productos AS producto ON detalle.id_producto = producto.id_producto
WHERE venta.id_venta = 34;

----Buscar detalles de inventario, incluyendo productos, proveedores, promociones y categorías:
SELECT producto.nombre AS producto, inventario.cantidad_disp, proveedor.nombre AS proveedor, promocion.tipo AS tipo_promocion, categoria.nombre AS categoria
FROM productos AS producto
JOIN inventario ON producto.id_inventario = inventario.id_inventario
JOIN proveedores AS proveedor ON producto.id_proveedor = proveedor.id_proveedor
JOIN promociones AS promocion ON producto.id_promocion = promocion.id_promocion
JOIN categorias AS categoria ON producto.id_categoria = categoria.id_categoria
WHERE producto.id_producto = 4;

94 %

Results Messages

	fecha_venta	total	cliente	comentario_cliente	producto	cantidad
1	2023-12-10	700	Cliente C	Buena atención al cliente, definitivamente volve...	Sofá	1

	producto	cantidad_disp	proveedor	tipo_promocion	categoria
1	Arroz	120	Proveedor D	Cupón de Regalo	Alimentos


```

--"¿Cuáles son los productos en inventario con una cantidad disponible superior a 50 unidades, incluyendo
--detalles sobre el nombre del producto, descripción, proveedor, categoría y la cantidad disponible?"

SELECT
    inventario.fecha_entrada,
    producto.nombre AS producto,
    producto.descripcion,
    proveedor.nombre AS proveedor,
    categorias.nombre AS categoria,
    inventario.cantidad_disp,
    promocion.tipo AS tipo_promocion,
    promocion.fecha_inicio,
    promocion.fecha_fin
FROM productos AS producto
JOIN proveedores AS proveedor ON producto.id_proveedor = proveedor.id_proveedor
JOIN categorias ON producto.id_categoria = categorias.id_categoria
JOIN inventario ON producto.id_inventario = inventario.id_inventario
JOIN promociones AS promocion ON producto.id_promocion = promocion.id_promocion
WHERE inventario.cantidad_disp > 50;

```

104 %

Results Messages

	fecha_entrada	producto	descripcion	proveedor	categoria	cantidad_disp	tipo_promocion	fecha_inicio	fecha_fin
1	2023-11-01	Smartphone	Último modelo con cámara de alta resolución	Proveedor A	Electrónicos	100	Descuento	2023-12-01	2023-12-15
2	2023-11-15	Camisa	Camisa de algodón de alta calidad	Proveedor B	Ropa	150	Oferta Especial	2023-12-05	2023-12-20
3	2023-11-30	Sofá	Sofá de tres plazas con diseño moderno	Proveedor C	Hogar	200	Rebaja de Temporada	2023-12-10	2023-12-31
4	2023-12-05	Arroz	Arroz de grano largo, paquete de 5kg	Proveedor D	Alimentos	120	Cupón de Regalo	2023-12-15	2023-12-25

```

--¿Podría proporcionar detalles sobre las ventas de productos electrónicos,
--incluyendo la fecha de venta, el nombre del producto y el nombre del cliente asociado?

SELECT
    v.fecha_venta,
    p.nombre AS producto,
    c.nombre AS cliente
FROM
    ventas v
INNER JOIN detalle_ventas d ON v.id_venta = d.id_venta
INNER JOIN productos p ON d.id_producto = p.id_producto
INNER JOIN clientes c ON v.id_cliente = c.id_cliente
INNER JOIN categorias cat ON p.id_categoria = cat.id_categoria

```

```

---- ¿Podría proporcionar información detallada sobre las ventas realizadas,
--incluyendo la fecha de venta, el total, la cantidad y el nombre del producto vendido, así como el nombre del cliente asociado?

SELECT
    p.nombre AS producto,
    p.descripcion,
    pr.nombre AS proveedor,
    c.nombre AS categoria,
    pm.tipo AS tipo_promocion
FROM
    productos p
INNER JOIN promociones pm ON p.id_promocion = pm.id_promocion
INNER JOIN proveedores pr ON p.id_proveedor = pr.id_proveedor
INNER JOIN categorias c ON p.id_categoria = c.id_categoria;

```

-----Funciones de Agregación = 4 Ejemplos simples-----

104 %

Results Messages Client Statistics

	producto	descripcion	proveedor	categoria	tipo_promocion
1	Smartphone	Último modelo con cámara de alta resolución	Proveedor A	Electrónicos	Descuento
2	Camisa	Camisa de algodón de alta calidad	Proveedor B	Ropa	Oferta Especial
3	Sofá	Sofá de tres plazas con diseño moderno	Proveedor C	Hogar	Rebaja de Temporada
4	Arroz	Arroz de grano largo, paquete de 5kg	Proveedor D	Alimentos	Cupón de Regalo


```

-----Funciones de Agregación = 4 Ejemplos simples-----
---1.¿Cuántos productos están actualmente en oferta?
SELECT COUNT(*) as productos_en_oferta
FROM productos
WHERE id_promoción IS NOT NULL;
---2.¿Cuántos pedidos se encuentran actualmente en proceso?
SELECT COUNT(*) as pedidos_en_proceso
FROM pedidos
WHERE estado = 'En Proceso';
---3.¿Cuál fue la venta más grande en términos de monto total?
SELECT MAX(total) as venta_mas_grande
FROM ventas;
---4.¿Cuál es el promedio de ventas diarias?
SELECT AVG(total) as promedio_ventas_diarias
FROM ventas;

--Funciones de Agregación II = 5 que maneje JOINS con mas de 3 tablas
--1.¿Cuántos productos diferentes se han vendido en cada categoría durante el último mes?
SELECT categorias.nombre AS categoria, COUNT(DISTINCT productos.id_producto) AS productos_vendidos

```

94 %

Results Messages Client Statistics

	productos_en_oferta
1	4

	pedidos_en_proceso
1	3

	venta_mas_grande
1	800

	promedio_ventas_diarias
1	390

```

---¿Cuáles son los detalles de los pedidos realizados, incluyendo el
--nombre del cliente, el estado del pedido y cualquier comentario asociado?
SELECT
    v.fecha_venta,
    v.total,
    d.cantidad,
    p.nombre AS producto,
    c.nombre AS cliente
FROM
    ventas v
INNER JOIN detalle_ventas d ON v.id_venta = d.id_venta
INNER JOIN productos p ON d.id_producto = p.id_producto
INNER JOIN clientes c ON v.id_cliente = c.id_cliente;

--- ¿Podría proporcionar información detallada sobre las ventas realizadas,
--incluyendo la fecha de venta, el total, la cantidad y el nombre del producto vendido, así como el nombre del cliente asociado?
SELECT
    p.nombre AS producto,

```

104 %

Results Messages

	fecha_venta	total	cantidad	producto	cliente
1	2023-12-01	800	2	Smartphone	Cliente A
2	2023-12-05	50	1	Camisa	Cliente B
3	2023-12-10	700	1	Sofá	Cliente C
4	2023-12-15	10	1	Arroz	Cliente D

```
--1.¿Cuántos productos diferentes se han vendido en cada categoría durante el último mes?
SELECT categorias.nombre AS categoria, COUNT(DISTINCT productos.id_producto) AS productos_vendidos
FROM categorias
JOIN productos ON categorias.id_categoria = productos.id_categoria
JOIN detalle_ventas ON productos.id_producto = detalle_ventas.id_producto
JOIN ventas ON detalle_ventas.id_venta = ventas.id_venta
WHERE ventas.fecha_venta BETWEEN '2023-11-01' AND '2023-12-01'
GROUP BY categorias.nombre;

--2.¿Cuál es el total de compras y el promedio de compras por cliente?
SELECT clientes.nombre AS cliente, COUNT(DISTINCT ventas.id_venta) AS total_compras, AVG(ventas.total) AS promedio_compras
FROM clientes
JOIN pedidos ON clientes.id_pedido = pedidos.id_pedido
JOIN ventas ON clientes.id_cliente = ventas.id_cliente
GROUP BY clientes.nombre;

--3. ¿Cuál es la cantidad total de productos en inventario suministrados por cada proveedor?
SELECT proveedores.nombre AS proveedor, SUM(inventario.cantidad_disp) AS total_en_inventario
FROM proveedores
JOIN productos ON proveedores.id_proveedor = productos.id_proveedor
JOIN inventario ON productos.id_inventario = inventario.id_inventario
GROUP BY proveedores.nombre;

-- 4.¿Cuál es la cantidad total y el precio promedio de productos en oferta por categoría?
SELECT categorias.nombre AS categoria, COUNT(productos.id_producto) AS total_ofertas, AVG(productos.precio) AS precio_promedio
```

94 %

Results Messages Client Statistics

categoria	productos_vendidos
1 Electrónicos	1

cliente	total_compras	promedio_compras
1 Cliente A	1	800
2 Cliente B	1	50
3 Cliente C	1	700
4 Cliente D	1	10

proveedor	total_en_inventario
1 Proveedor A	100
2 Proveedor B	150
3 Proveedor C	200
4 Proveedor D	120

categoria	total_ofertas	precio_promedio
1 Alimentos	1	10
2 Electrónicos	1	800

<https://drive.google.com/file/d/1eeffqZYuORB6K7uAiwKRF9hOnwfvZYm7/view?usp=sharing>

4. CONCLUSIONES

1. Logro de Objetivos Principales:

El proyecto logra satisfactoriamente su objetivo principal de desarrollar un sistema eficiente de gestión de productos. La base de datos está estructurada de manera coherente y proporciona una sólida base para mejorar la experiencia de los usuarios al buscar y adquirir productos específicos.

2. Mejoras Potenciales en Recomendaciones Personalizadas:

Aunque la base de datos permite la integración de datos para recomendaciones personalizadas, sería beneficioso explorar y desarrollar algoritmos más avanzados que aprovechen al máximo la información disponible. Esto podría mejorar aún más la precisión de las sugerencias personalizadas.

3. Eficiencia en la Gestión de Inventarios y Proveedores:

La inclusión de tablas como "inventario" y "proveedores" demuestra una eficiencia en la gestión de existencias y relaciones con proveedores. Sin embargo, para optimizar aún más este aspecto, se podría considerar la implementación de notificaciones automáticas de reabastecimiento o herramientas de análisis predictivo.

4. Seguimiento Detallado de Ventas para Estrategias Comerciales:

El seguimiento detallado de pedidos y ventas proporciona una sólida base para la evaluación del rendimiento. Se sugiere implementar análisis más avanzados para identificar patrones de compra, evaluar la efectividad de promociones y ajustar estrategias comerciales de manera proactiva.

5. Explorar Oportunidades para Mejorar la Logística y Entrega:

La inclusión de información sobre la ubicación del cliente abre oportunidades para mejorar la logística y la entrega. Sería beneficioso explorar la implementación de herramientas de optimización de rutas y sistemas de seguimiento en tiempo real para mejorar la eficiencia en la entrega de productos.