

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”

EL ALTO 11-12-23

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.

1. 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.



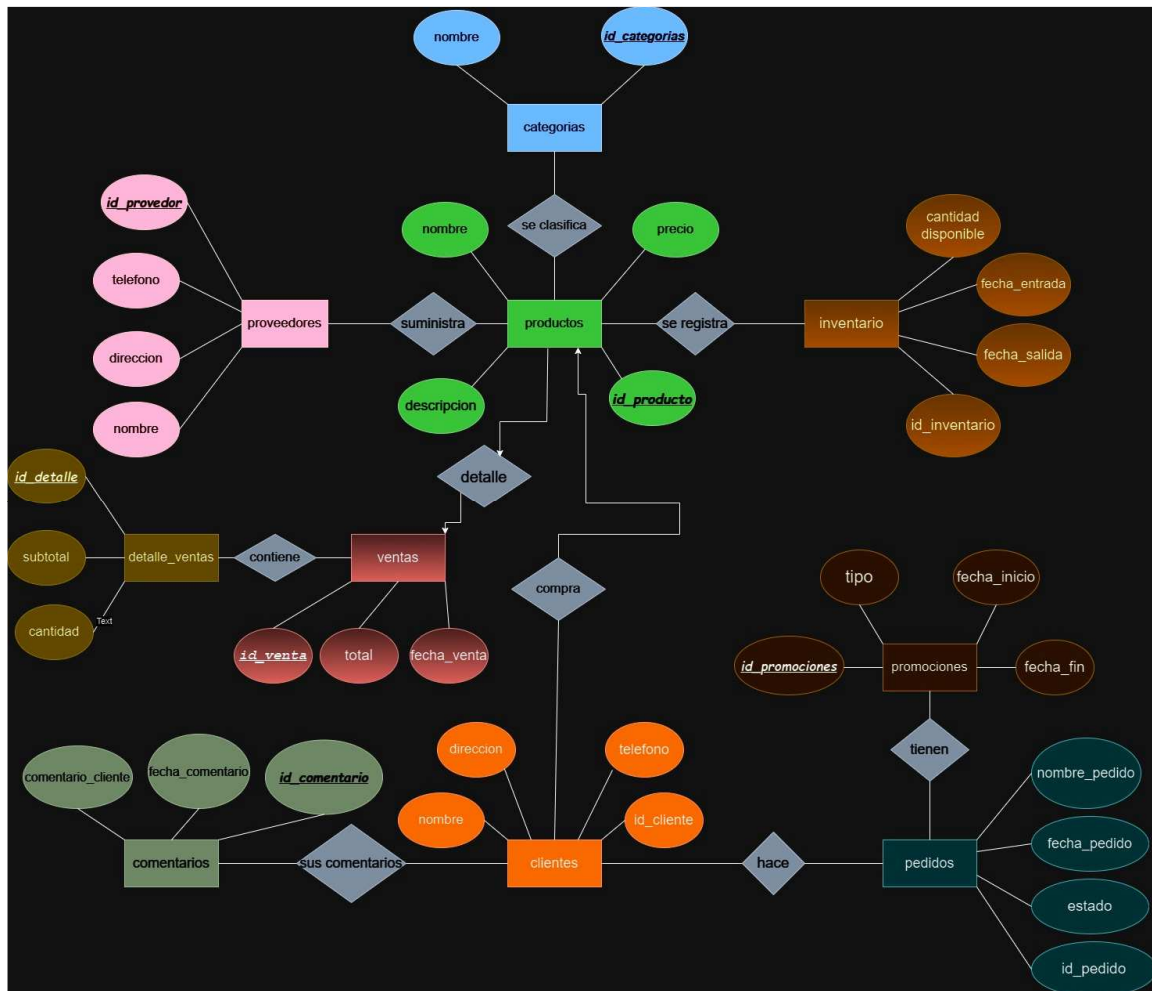
b. Entidades/tablas de sistema: Las tablas necesarias que se identificaron

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.

Diseño E-R:



<https://drive.google.com/file/d/1pEbkP8sGVwaJnrzawoljleK98GN-BPHc/view?usp=sharing>

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 ('videojuegos', 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 ('play 5', 'Frecuencia variable de GPU hasta 2.23 GHz (10.3 TFLOPS)', 10,  
'videojuegos', 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,
    telefonno 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, telefonno, id_pedido, id_comentario, id_cliente)
VALUES ('Cliente A', '456 Calle Principal', 123456, 1, 1, 11);

INSERT INTO clientes (nombre, direccion, telefonno, id_pedido, id_comentario, id_cliente)
VALUES ('Cliente B', '789 Avenida Secundaria', 98765, 2, 2, 22);

INSERT INTO clientes (nombre, direccion, telefonno, id_pedido, id_comentario, id_cliente)
VALUES ('Cliente C', '012 Carretera Central', 246810, 3, 3, 33);

INSERT INTO clientes (nombre, direccion, telefonno, 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 son los nombres de los productos, nombres de categorías,

--precios de los productos, nombres de proveedores y direcciones de proveedores?

```
SELECT productos.nombre,  
       categorias.nombre,  
       productos.precio,  
       proveedores.nombre,  
       proveedores.direccion  
FROM productos  
INNER JOIN categorias ON productos.id_categoria = categorias.id_categoria  
INNER JOIN proveedores ON productos.id_proveedor = proveedores.id_proveedor  
INNER JOIN detalle_ventas ON productos.id_producto = detalle_ventas.id_producto  
INNER JOIN ventas ON detalle_ventas.id_venta = ventas.id_venta;
```

---2. ¿Cuál es la fecha de venta, nombre del producto y nombre del cliente
--para las ventas de productos de la categoría 'Electrónicos'?

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

---3. ¿Cuáles son los nombres de los productos, tipos de promociones,
--nombres de proveedores, nombres de categorías y cantidades disponibles en
inventario para los productos?

```
SELECT productos.nombre,
       promociones.tipo,
       proveedores.nombre,
       categorias.nombre,
       inventario.cantidad_disp
FROM productos
JOIN promociones ON productos.id_promocion = promociones.id_promocion
JOIN proveedores ON productos.id_proveedor = proveedores.id_proveedor
JOIN categorias ON productos.id_categoria = categorias.id_categoria
JOIN inventario ON productos.id_inventario = inventario.id_inventario;
```

---4. ¿Cuáles son los nombres de los pedidos, estados de los pedidos, nombres de
clientes,
--nombres de productos y fechas de venta?

```
SELECT pedidos.nombre_pedido,
       pedidos.estado,
       clientes.nombre,
       productos.nombre,
       ventas.fecha_venta
FROM pedidos
JOIN clientes ON pedidos.id_pedido = clientes.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;
```

---5. ¿Cuáles son las fechas de venta, nombres de clientes
--y nombres de proveedores asociados a las ventas donde el proveedor es 'Proveedor
A'?

```
SELECT v.fecha_venta, c.nombre, pr.nombre
FROM ventas v
JOIN clientes c ON v.id_cliente = c.id_cliente
JOIN detalle_ventas dv ON v.id_venta = dv.id_venta
JOIN productos p ON dv.id_producto = p.id_producto
JOIN proveedores pr ON p.id_proveedor = pr.id_proveedor
WHERE pr.nombre = 'Proveedor A';
```

--3 JOINS CON MAS DE 2 TABLAS--

--1. ¿Cuáles son los nombres de los productos, tipos de promociones,
--nombres de proveedores, nombres de categorías y cantidades disponibles en
inventario para los productos?

```
SELECT productos.nombre,  
       promociones.tipo,  
       proveedores.nombre,  
       categorias.nombre,  
       inventario.cantidad_disp  
FROM productos  
JOIN promociones ON productos.id_promocion = promociones.id_promocion  
JOIN proveedores ON productos.id_proveedor = proveedores.id_proveedor  
JOIN categorias ON productos.id_categoria = categorias.id_categoria  
JOIN inventario ON productos.id_inventario = inventario.id_inventario;
```

--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;
```

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

```
SELECT COUNT(*) AS total_pedidos FROM pedidos;
```

```
SELECT SUM(precio) AS total_ventas FROM productos;
```

```
SELECT AVG(precio) AS promedio_precio FROM productos;
```

```
SELECT MAX(precio) AS precio_maximo FROM productos;
```

```
SELECT MIN(precio) AS precio_minimo FROM productos;
```

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

-- 1.¿Cuántos productos hay en cada categoría y cuál es el nombre de cada categoría?

```
SELECT categorias.nombre AS categoria, COUNT(productos.id_producto) AS
cantidad_productos
FROM categorias
INNER JOIN productos ON categorias.id_categoria = productos.id_categoria
INNER JOIN inventario ON productos.id_inventario = inventario.id_inventario
GROUP BY categorias.nombre;
```

--2. --¿Cuál fue el total de ventas realizadas por cliente en cada fecha de venta?

```
SELECT ventas.fecha_venta, clientes.nombre, SUM(detalle_ventas.subtotal) AS
total_venta
FROM ventas
INNER JOIN clientes ON ventas.id_cliente = clientes.id_cliente
INNER JOIN detalle_ventas ON ventas.id_venta = detalle_ventas.id_venta
GROUP BY ventas.fecha_venta, clientes.nombre;
```

--3.¿Cuál es el total de ventas por categoría de productos

--, considerando las ventas totales de cada producto dentro de esa categoría?

```
SELECT c.nombre, SUM(v.total) as total_ventas
FROM ventas v
INNER JOIN clientes cl ON v.id_cliente = cl.id_cliente
INNER JOIN detalle_ventas dv ON v.id_venta = dv.id_venta
INNER JOIN productos p ON dv.id_producto = p.id_producto
INNER JOIN categorias c ON p.id_categoria = c.id_categoria
GROUP BY c.nombre;
```

--4. ¿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;
```

--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:



2. USABILIDAD

Results Messages

	tipo	fecha_inicio	fecha_fin	id_promocion					
1	Descuento	2023-12-01	2023-12-15	1					
2	Oferta Especial	2023-12-05	2023-12-20	2					
3	Rebaja de Temporada	2023-12-10	2023-12-31	3					
4	Cupón de Regalo	2023-12-15	2023-12-25	4					

	nombre	direccion	telefono	id_proveedor					
1	Proveedor A	123 Calle Principal	123456	1					
2	Proveedor B	456 Avenida Secundaria	987654	2					
3	Proveedor C	789 Carretera Central	246810	3					
4	Proveedor D	012 Calle Secundaria	36912...	4					

	nombre	id_categoria							
1	Electrónicos	1							
2	Ropa	2							
3	Hogar	3							
4	videojuegos	4							

	cantidad_disp	fecha_entrada	fecha_salida	id_inventario					
1	100	2023-11-01	2023-12-01	1					
2	150	2023-11-15	2023-12-15	2					
3	200	2023-11-30	2023-12-30	3					
4	120	2023-12-05	2023-12-31	4					

	nombre	descripcion	precio	categoria	id_promocion	id_proveedor	id_categoria	id_inventario	id_producto
1	Smartphone	Último modelo con cámara de alta resolución	800	Electrónicos	1	1	1	1	1
2	Camisa	Camisa de algodón de alta calidad	50	Ropa	2	2	2	2	2
3	Sofá	Sofá de tres plazas con diseño moderno	700	Hogar	3	3	3	3	3
4	play 5	Frecuencia variable de GPU hasta 2.23 G...	10	videojuegos	4	4	4	4	4

	nombre_pedido	fecha_pedido	estado	id_pedido					
1	Pedido 1	2023-12-01	En Proceso	1					
2	Pedido 2	2023-12-05	En Proceso	2					
3	Pedido 3	2023-12-10	Entregado	3					
4	Pedido 4	2023-12-15	En Proceso	4					

	comentario_cliente	fecha_comentario	id_comentario			
1	Excelente servicio, muy satisfecho con mi compra.	2023-12-01	1			
2	El producto llegó en mal estado, espero una pron...	2023-12-05	2			
3	Buena atención al cliente, definitivamente volver...	2023-12-10	3			
4	El envío se retrasó, necesito una explicación.	2023-12-15	4			

	nombre	direccion	telefono	id_pedido	id_comentario	id_cliente			
1	Juan Matinez	456 Calle Principal	123456	1	1	11			
2	pedro Hemande	789 Avenida Secundaria	98765	2	2	22			
3	jose Perez	012 Carretera Central	246810	3	3	33			
4	femando Cond...	123 Calle Secundaria	36912	4	4	44			

	fecha_venta	total	id_cliente	id_venta				
1	2023-12-01	800	11	12				
2	2023-12-05	50	22	23				
3	2023-12-10	700	33	34				
4	2023-12-15	10	44	45				

	cantidad	subtotal	id_producto	id_venta	id_detalle				
1	2	1600	1	12	111				
2	1	50	2	23	222				
3	1	700	3	34	333				
4	1	10	4	45	444				

```

---4 Joins con más de 5 tablas:
--¿Cuáles son los nombres de los productos, nombres de categorías,
--precios de los productos, nombres de proveedores y direcciones de proveedores?

SELECT productos.nombre,
       categorias.nombre,
       productos.precio,
       proveedores.nombre,
       proveedores.direccion
FROM productos
INNER JOIN categorias ON productos.id_categoria = categorias.id_categoria
INNER JOIN proveedores ON productos.id_proveedor = proveedores.id_proveedor
INNER JOIN detalle_ventas ON productos.id_producto = detalle_ventas.id_producto
INNER JOIN ventas ON detalle_ventas.id_venta = ventas.id_venta;

```

100 %

Results Messages

	nombre	nombre	precio	nombre	direccion
1	Smartphone	Electrónicos	800	Proveedor A	123 Calle Principal
2	Camisa	Ropa	50	Proveedor B	456 Avenida Secundaria
3	Sofá	Hogar	700	Proveedor C	789 Carretera Central
4	play 5	videojuegos	10	Proveedor D	012 Calle Secundaria

```

--¿Cuáles fueron las fechas de las ventas realizadas por nombre de producto,
--cantidad, total de venta y nombre del cliente?

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;

```

100 %

Results Messages

	fecha_venta	total	cantidad	producto	cliente
1	2023-12-01	800	2	Smartphone	juan Matinez
2	2023-12-05	50	1	Camisa	pedro Hemande
3	2023-12-10	700	1	Sofá	jose Perez
4	2023-12-15	10	1	play 5	femando Condori

```

----¿Cuál es la fecha de venta, nombre del producto y nombre del cliente
--para las ventas de productos de la categoría 'Electrónicos'?

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

```

%

Results Messages

fecha_venta	producto	cliente
2023-12-01	Smartphone	juan Matinez

```
--¿Cuáles son los nombres de los productos, su descripciones del producto,
--nombre de los proveedores el nombre de la categorías y tipos de promoción?
```

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

proveedor	producto	descripcion	categoria	tipo
Proveedor A	Smartphone	Último modelo con cámara de alta resolución	Electrónicos	Descuento
Proveedor B	Camisa	Camisa de algodón de alta calidad	Ropa	Oferta Especial
Proveedor C	Sofá	Sofá de tres plazas con diseño moderno	Hogar	Rebaja de Temporada
Proveedor D	play 5	Frecuencia variable de GPU hasta 2.23 GHz (10.3 ...	videojuegos	Cupón de Regalo

```
--¿Cuáles son los nombres de los productos, tipos de promociones,
--nombres de proveedores, nombres de categorías y cantidades disponibles en inventario para los productos?
```

```
SELECT productos.nombre,
    promociones.tipo,
    proveedores.nombre,
    categorias.nombre,
    inventario.cantidad_disp
FROM productos
JOIN promociones ON productos.id_promocion = promociones.id_promocion
JOIN proveedores ON productos.id_proveedor = proveedores.id_proveedor
JOIN categorias ON productos.id_categoria = categorias.id_categoria
JOIN inventario ON productos.id_inventario = inventario.id_inventario;
```

nombre	tipo	nombre	nombre	cantidad_disp
Smartphone	Descuento	Proveedor A	Electrónicos	100
Camisa	Oferta Especial	Proveedor B	Ropa	150
Sofá	Rebaja de Temporada	Proveedor C	Hogar	200
play 5	Cupón de Regalo	Proveedor D	videojuegos	120

```
--¿Cuáles son los nombres de los pedidos, estados de los pedidos, nombres de clientes,
--nombres de productos y fechas de venta?
```

```
SELECT pedidos.nombre_pedido,
    pedidos.estado,
    clientes.nombre,
    productos.nombre,
    ventas.fecha_venta
FROM pedidos
JOIN clientes ON pedidos.id_pedido = clientes.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;
```

nombre_pedido	estado	nombre	nombre	fecha_venta
Pedido 1	En Proceso	juan Matinez	Smartphone	2023-12-01
Pedido 2	En Proceso	pedro Hemande	Camisa	2023-12-05
Pedido 3	Entregado	jose Perez	Sofá	2023-12-10
Pedido 4	En Proceso	fernando Condori	play 5	2023-12-15

```
--¿Cuáles son las fechas de venta, nombres de clientes
--y nombres de proveedores asociados a las ventas donde el proveedor es 'Proveedor A'
```

```
SELECT v.fecha_venta, c.nombre, pr.nombre
FROM ventas v
JOIN clientes c ON v.id_cliente = c.id_cliente
JOIN detalle_ventas dv ON v.id_venta = dv.id_venta
JOIN productos p ON dv.id_producto = p.id_producto
JOIN proveedores pr ON p.id_proveedor = pr.id_proveedor
WHERE pr.nombre = 'Proveedor A';
```

fecha_venta	nombre	nombre
2023-12-01	Juan Matinez	Proveedor A

```
---Funciones de agregacion
```

```
SELECT COUNT(*) AS total_pedidos FROM pedidos;
SELECT SUM(precio) AS total_ventas FROM productos;
SELECT AVG(total) AS promedio_ventas_diarias FROM ventas;
SELECT MAX(total) AS venta_mas_grande FROM ventas;
SELECT MIN(precio) AS precio_minimo FROM productos;
```

total_pedidos
4
total_ventas
1560
promedio_ventas_diarias
390
venta_mas_grande
800
precio_minimo
10

```
----Funciones de Agregación con inner join
```

```
--¿Cuántos productos hay en cada categoría y cuál es el nombre de cada categoría?
```

```
SELECT categorias.nombre AS categoria, COUNT(productos.id_producto) AS cantidad_productos
FROM categorias
INNER JOIN productos ON categorias.id_categoria = productos.id_categoria
INNER JOIN inventario ON productos.id_inventario = inventario.id_inventario
GROUP BY categorias.nombre;
```

categoria	cantidad_productos
Electrónicos	1
Hogar	1
Ropa	1
videojuegos	1

```
--¿Cuál es el promedio de ventas para cada categoría de productos |
--considerando las ventas totales de cada producto dentro de esa categoría?
```

```
SELECT categorias.nombre AS categoria,
       AVG(ventas.total) AS promedio_venta
FROM categorias
INNER JOIN productos ON categorias.id_categoria = productos.id_categoria
INNER JOIN detalle_ventas ON productos.id_producto = detalle_ventas.id_producto
INNER JOIN ventas ON detalle_ventas.id_venta = ventas.id_venta
GROUP BY categorias.nombre;
```

Results Messages

categoria	promedio_venta
Electrónicos	800
Hogar	700
Ropa	50
videojuegos	10

```
--¿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;
```

Results Messages

cliente	total_compras	promedio_compras
femando Condori	1	10
jose Perez	1	700
juan Matinez	1	800
pedro Hemande	1	50

```
--¿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;
```

00 %

Results Messages

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

--¿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;			
00 %			
Results Messages			
	cliente	proveedor	total_comprado
1	Juan Matinez	Proveedor A	800
2	pedro Hemande	Proveedor B	50
3	jose Perez	Proveedor C	700
4	fernando Condori	Proveedor D	10

https://drive.google.com/file/d/1czomq_xkvz7VLRyZn4cciW4XxSJlXGIR/view?usp=sharing

3. 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.