

Proyecto DB SQL

Introducción.....	3
Descripción de la Situación del Negocio	3
Objetivos de la Base de Datos	3
Problemas a resolver	4
Entidades y relaciones	5
Diagrama Entidad-Relación	5
Descripción de tablas	6
Tabla: usuarios	6
Tabla: historial_compras	6
Tabla: categorias	7
Tabla: productos	7
Tabla: detalle_historial	7
Tabla: proveedores	8
Tabla: promociones	8
Tabla: ordenes	9
Tabla: ordenes_promociones	9
Tabla: envios	9
Tabla: detalle_orden	10
Tabla: carrito	10
Tabla: detalle_carrito	11
Tabla: comentarios	11

Scripts	12
Creación de tablas	12
Inserción de datos.....	18
Vistas Principales.....	22
Funciones	27
Stored Procedures	29
Triggers	32
Definición de usuarios y permisos.....	38
Transacciones	39
Reportes.....	43
Herramientas Utilizadas	44

Proyecto DB SQL

Introducción

Descripción de la Situación del Negocio

En el núcleo de la creación de nuestro Marketplace yace la esencial tarea de diseñar y construir una base de datos robusta que permitirá respaldar la gestión eficiente de productos, la personalización de la experiencia del usuario, la seguridad de datos y un rendimiento óptimo. En este proceso, nos embarcamos en la tarea de analizar requisitos específicos, diseñar la estructura, implementar con un enfoque escalable. Con cada paso, nuestro objetivo es forjar una base de datos que no solo cumpla con las necesidades actuales de nuestro Marketplace, sino que también sienta las bases para su crecimiento sostenible y la entrega de una experiencia excepcional a nuestros usuarios y socios comerciales.

Objetivos de la Base de Datos

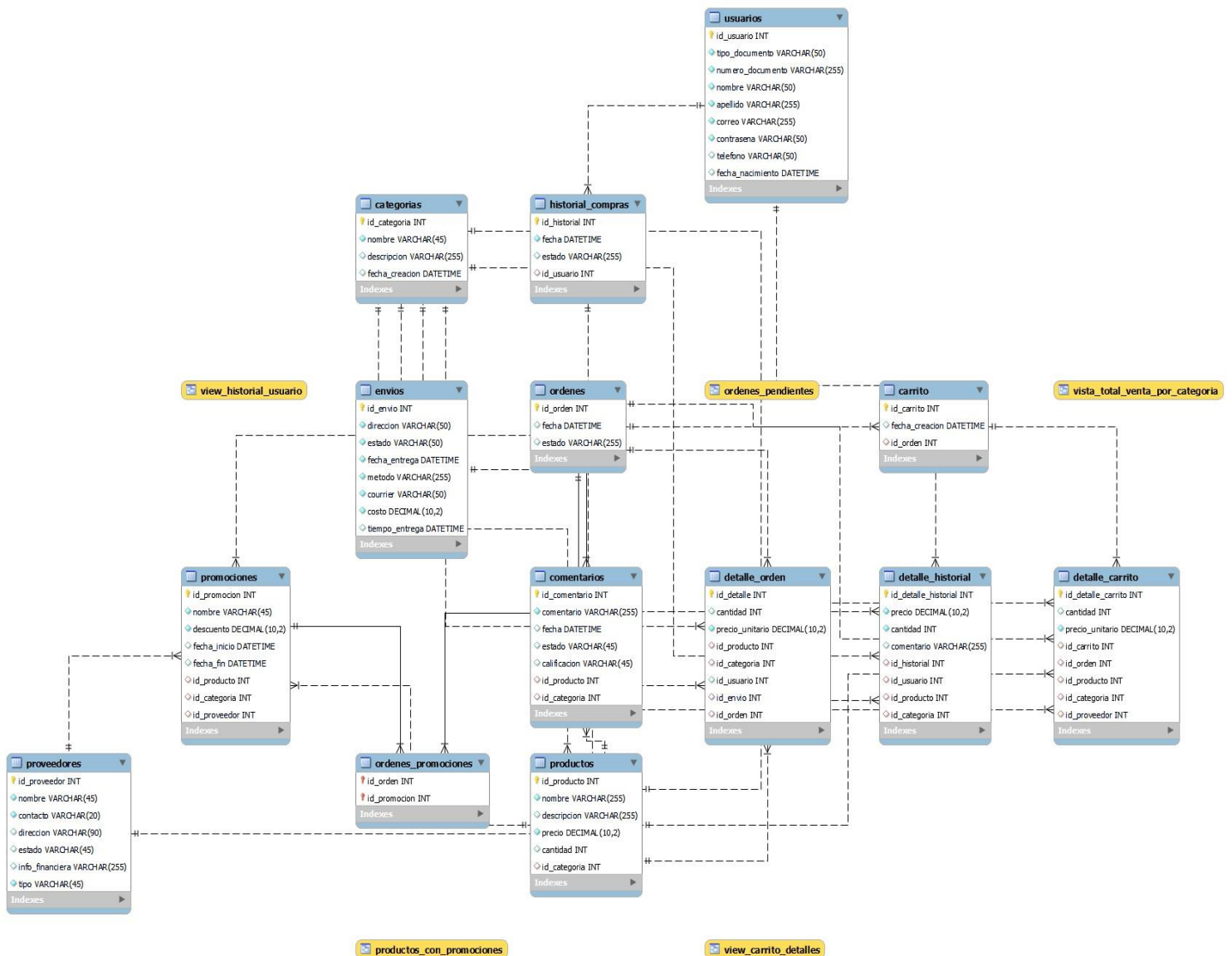
- Almacenar información detallada sobre cada producto, como nombre, descripción, precio, imágenes, categoría, atributos específicos, etc.
- Mantener un seguimiento actualizado del inventario para evitar problemas de sobreventa o falta de existencias.
- Almacenar datos de usuarios registrados, incluyendo nombres, direcciones, información de contacto y preferencias.
- Registrar información financiera relacionada con las transacciones, como precios, impuestos, descuentos, métodos de pago y estado de pago.

Problemas a resolver

1. **Gestión de Inventario:** Controlar y actualizar en tiempo real el inventario de productos. La base de datos permitirá almacenar información sobre la disponibilidad de productos, evitando problemas como la sobreventa o la falta de existencias.
2. **Seguimiento de Transacciones:** Gestionar y realizar un seguimiento de las transacciones de compra. La base de datos permitirá almacenar datos de transacciones, incluyendo detalles financieros y estados de pago, para facilitar el seguimiento y la resolución de problemas relacionados con los pedidos.
3. **Personalización de Experiencia:** Ofrecer una experiencia personalizada a los usuarios. La base de datos permitirá almacenar historiales de compras y preferencias de usuarios, lo que permite personalizar recomendaciones y promociones.
4. **Eficiencia en Búsquedas y Filtrados:** Realizar búsquedas y filtrados eficientes en un catálogo extenso. La base de datos permitirá utilizar índices y consultas eficientes en la base de datos para mejorar la velocidad de recuperación de datos y proporcionar una experiencia de usuario fluida.
5. **Seguridad y Privacidad:** Garantizar la seguridad y privacidad de los datos almacenados. La base de datos permitirá implementar medidas de seguridad en la base de datos para proteger la integridad y confidencialidad de la información, así como gestionar los permisos de acceso de los usuarios.
6. **Escalabilidad:** Manejar el aumento en el volumen de datos y transacciones. La base de datos permitirá diseñar la base de datos de manera escalable para adaptarse al crecimiento del Marketplace sin comprometer el rendimiento.

Entidades y relaciones

Diagrama Entidad-Relación



Descripción de tablas

Tabla: usuarios

Esta tabla almacena información sobre los usuarios registrados en el marketplace.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_usuario	INT	Identificador único del usuario.
	tipo_documento	VARCHAR (45)	Tipo de documento de identidad.
	numero_documento	VARCHAR (45)	Número documento de identidad.
	nombre	VARCHAR (45)	Nombre completo del usuario.
	apellido	VARCHAR (45)	Apellido completo del usuario.
	correo	VARCHAR (80)	Correo electrónico del usuario
	contrasena	VARCHAR (45)	Contraseña del usuario.
	telefono	VARCHAR (45)	Teléfono del usuario.
	fecha_nacimiento	DATETIME	Fecha de nacimiento.

Tabla: historial_compras

Esta tabla almacena las compras realizadas de los usuarios.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_historial	INT	Identificador único del historial de compras
FK	id_usuario	INT	ID del usuario asociado.
	fecha	DATETIME	Calle y número de la dirección.
	estado	VARCHAR (20)	Ciudad de la dirección.

Tabla: categorias

Esta tabla almacena información sobre los productos disponibles en el marketplace.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_categoria	INT	Identificador único de la categoría del producto.
	nombre	INT	Nombre de la categoría del producto.
	descripcion	INT	Descripción de la categoría.
	fecha_creacion	DATETIME	Fecha de Creación de una nueva categoría.

Tabla: productos

Esta tabla almacena información sobre los productos disponibles en el marketplace.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_producto	INT	Identificador único del producto.
FK	id_categoria	INT	ID de la categoría del producto asociado.
	nombre	VARCHAR (45)	Nombre del producto
	descripcion	VARCHAR (45)	Descripción detalla del producto.
	precio	DECIMAL (10, 2)	Precio de cada producto.
	cantidad	INT	Cantidad en stock de cada producto.

Tabla: detalle_historial

Esta tabla almacena información sobre los usuarios registrados en el marketplace.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_detalle_historial	INT	Identificador único del detalle del historial.
FK	id_historial	INT	ID del historial de compras asociado.
FK	id_usuario	INT	ID del usuario asociado.
FK	id_producto	INT	ID del producto asociado.
FK	id_categoria	INT	ID de la categoría del producto asociado.
	cantidad	INT	Teléfono del usuario.
	precio	DECIMAL (10, 2)	Precio de cada producto.
	comentario	VARCHAR (45)	Comentarios detalles del historial.

Tabla: proveedores

Esta tabla almacena información de los proveedores fuente importante en el suministro de productos y servicios.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_proveedor	INT	Identificador único del proveedor.
	nombre	VARCHAR (45)	Nombre del proveedor.
	contacto	VARCHAR (45)	Número fijo o móvil del proveedor.
	direccion	VARCHAR (45)	Ubicación de la empresa.
	estado	VARCHAR (45)	Estado del proveedor: activo, inactivo, suspendido.
	info_financiera	VARCHAR (45)	Datos financieros relevantes.
	tipo	VARCHAR (45)	Indica el tipo: bienes o servicios.

Tabla: promociones

Esta tabla almacena información sobre las promociones o descuentos que la empresa ofrece a sus clientes.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_promocion	INT	Identificador único de la promoción.
FK	id_producto	INT	ID del producto asociado.
FK	id_categoria	INT	ID de la categoría asociada.
FK	id_proveedor	INT	ID del proveedor asociado.
	nombre	VARCHAR (45)	Nombre de la promoción.
	descuento	DECIMAL (10, 2)	Descuento aplicado a la promoción.
	fecha_inicio	DATETIME	Fecha inicial de la promoción.
	fecha_fin	DATETIME	Fecha máxima de la promoción.

Tabla: ordenes

Esta tabla almacena gestión de pedidos sobre cada transacción de compra.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_orden	INT	Identificador único por cada artículo en el carrito.
	fecha	DATETIME	Fecha de creación de la orden.
	estado	VARCHAR (45)	Estado de la orden creada.

Tabla: ordenes_promociones

Esta tabla almacena información sobre las promociones por cada orden.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_orden	INT	Identificador único del comentario.
FK	id_promocion	INT	ID del producto asociado.

Tabla: envios

Esta tabla almacena información de los envíos de cualquier producto solicitado por el usuario.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_envio	INT	Identificador único del envío.
	direccion	VARCHAR (45)	Dirección donde será entregado el producto.
	estado	VARCHAR (45)	Estado del envío: en tránsito, entregado, pendiente.
	Fecha_entrega	DATETIME	Fecha de entrega del producto.
	metodo	VARCHAR (45)	Método de envío: estándar, express, recogido en tienda.
	courrier	VARCHAR (45)	Datos contacto empresa mensajera.
	costo	DECIMAL (10,2)	Costo por el transporte.
	tiempo_entrega	DATETIME	Tiempo en minutos tomará entrega del producto.

Tabla: detalle_orden

Esta tabla almacena información detallada de las transacciones de gestión de pedidos.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_detalle	INT	Identificador único por cada artículo de la orden.
FK	id_producto	INT	ID del producto asociado.
FK	id_categoria	INT	ID de la categoría asociado.
FK	id_envio	INT	ID del envío asociado.
FK	id_orden	INT	ID del orden asociado.
	cantidad	INT	Cantidad de unidades del producto incluido en la orden.
	precio_unitario	DECIMAL (10,2)	Costo total del artículo.

Tabla: carrito

Esta tabla almacena información de los envíos de cualquier producto solicitado por el usuario.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_carrito	INT	Identificador único por cada carrito de compras.
KF	id_orden	INT	ID de la orden asociada.
	fecha_entrega	DATETIME	Fecha y hora que se creó el carrito.

Tabla: detalle_carrito

Esta tabla almacena información detalla del carrito de compras.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_detalle_carrito	INT	Identificador único por cada artículo en el carrito.
FK	id_carrito	INT	ID del usuario asociado.
FK	id_orden	INT	ID del producto asociado.
FK	id_producto	INT	ID del producto asociado.
FK	id_categoria	INT	ID de la categoría asociada.
FK	id_proveedor	INT	ID del proveedor asociado.
	cantidad	INT	Cantidad de productos.
	precio_unitario	DECIMAL (10,2)	Precio total del producto.

Tabla: comentarios

Esta tabla almacena información sobre los comentarios de cada usuario por cada producto.

Tipo de Clave	Nombre del Campo	Tipo de Datos	Descripción
PK	id_comentario	INT	Identificador único del comentario.
FK	id_producto	INT	ID del producto asociado.
FK	id_categoria	INT	ID de la categoría asociada.
	comentario	VARCHAR (45)	Contenido real del comentario.
	fecha	TIMESTAMP	Tiempo que se hizo el comentario.
	estado	VARCHAR (45)	Estado del comentario.
	calificacion	VARCHAR (45)	Calificación o puntaje de los comentarios.

Scripts

Creación de tablas

```
-- Crear la base de datos del marketplace
CREATE DATABASE IF NOT EXISTS marketplace_db;

USE marketplace_db;

-- Tabla de Usuarios
CREATE TABLE IF NOT EXISTS usuarios
(
    id_usuario      INT auto_increment PRIMARY KEY,
    tipo_documento  VARCHAR(50) NOT NULL,
    dni             INT NOT NULL UNIQUE,
    nombre          VARCHAR(50) NOT NULL,
    apellido        VARCHAR(255) NOT NULL,
    correo          VARCHAR(255) NOT NULL UNIQUE,
    contrasena      VARCHAR(50) NOT NULL,
    telefono        VARCHAR(50),
    fecha_nacimiento DATETIME
);

-- Tabla de Historial de compras
CREATE TABLE IF NOT EXISTS historial_compras
(
    id_historial INT auto_increment PRIMARY KEY,
    fecha        DATETIME NOT NULL,
    estado       VARCHAR(255),
    id_usuario   INT,
    FOREIGN KEY(id_usuario) REFERENCES usuarios(id_usuario)
);

-- Tabla de Categorías
```

```

CREATE TABLE IF NOT EXISTS categorias
(
    id_categoria    INT auto_increment PRIMARY KEY,
    nombre          VARCHAR(45) NOT NULL,
    descripcion     VARCHAR(255),
    fecha_creacion  DATETIME
);

-- Tabla de Productos
CREATE TABLE IF NOT EXISTS productos
(
    id_producto    INT auto_increment PRIMARY KEY,
    nombre         VARCHAR(255) NOT NULL,
    description     VARCHAR(255),
    precio         DECIMAL(10, 2) NOT NULL,
    cantidad       INT
);

-- Tabla del Detalle del historial de compras
CREATE TABLE IF NOT EXISTS detalle_historial
(
    id_detalle_historial INT auto_increment PRIMARY KEY,
    precio               DECIMAL(10, 2) NOT NULL,
    cantidad             INT NOT NULL,
    comentario          VARCHAR(255),
    id_historial         INT,
    id_usuario           INT,
    id_producto          INT,
    id_categoria         INT,
    FOREIGN KEY(id_historial) REFERENCES historial_compras(id_historial),
    FOREIGN KEY(id_usuario) REFERENCES usuarios(id_usuario),
    FOREIGN KEY(id_categoria) REFERENCES categorias(id_categoria),
    FOREIGN KEY(id_producto) REFERENCES productos(id_producto)
);

-- Tabla de Proveedores

```

```
CREATE TABLE IF NOT EXISTS proveedores
```

```
(
    id_proveedor    INT auto_increment PRIMARY KEY,
    nombre          VARCHAR(45) NOT NULL,
    contacto        DECIMAL(10, 2) NOT NULL,
    direccion       VARCHAR(90),
    estado          VARCHAR(45),
    info_financiera VARCHAR(255),
    tipo            VARCHAR(45) NOT NULL
);
```

```
-- Tabla de Promociones
```

```
CREATE TABLE IF NOT EXISTS promociones
```

```
(
    id_promocion INT auto_increment PRIMARY KEY,
    nombre       VARCHAR(45) NOT NULL,
    descuento    DECIMAL(10, 2) NOT NULL,
    fecha_inicio DATETIME,
    fecha_fin    DATETIME,
    id_producto  INT,
    id_categoria INT,
    id_proveedor INT,
    FOREIGN KEY(id_producto) REFERENCES productos(id_producto)
),
    FOREIGN KEY(id_categoria) REFERENCES categorias(id_categoria),
    FOREIGN KEY(id_proveedor) REFERENCES proveedores(id_proveedor)
);
```

```
-- Tabla de Ordenes
```

```
CREATE TABLE IF NOT EXISTS ordenes
```

```
(
    id_orden      INT auto_increment PRIMARY KEY,
    fecha         DATETIME,
    estado        VARCHAR(255),
    id_promocion  INT,
    FOREIGN KEY(id_promocion) REFERENCES promociones(id_promo
cion)
);
```

-- Tabla de Envíos

```
CREATE TABLE IF NOT EXISTS envios
```

```
(
    id_envio      INT auto_increment PRIMARY KEY,
    direccion     VARCHAR(50) NOT NULL,
    estado        VARCHAR(50) NOT NULL,
    fecha_entrega DATETIME NOT NULL,
    metodo        VARCHAR(255) NOT NULL UNIQUE,
    courier       VARCHAR(50) NOT NULL,
    costo         DECIMAL(10, 2) NOT NULL,
    tiempo_entrega DATETIME
);
```

-- Tabla Detalle de la Orden

```
CREATE TABLE IF NOT EXISTS detalle_orden
```

```
(
    id_detalle    INT auto_increment PRIMARY KEY,
    cantidad      INT,
    precio_total  DECIMAL(10, 2) NOT NULL,
    id_producto   INT,
    id_categoria  INT,
    id_usuario    INT,
    id_envio      INT,
    id_orden      INT,
    FOREIGN KEY(id_producto) REFERENCES productos(id_producto
),
```

```

        FOREIGN KEY(id_categoria) REFERENCES categorias(id_categoria),
        FOREIGN KEY(id_envio) REFERENCES envios(id_envio),
        FOREIGN KEY(id_orden) REFERENCES ordenes(id_orden)
    );

```

-- Tabla de Carrito de compras

```

CREATE TABLE IF NOT EXISTS carrito
(
    id_carrito      INT auto_increment PRIMARY KEY,
    fecha_creacion  DATETIME,
    id_orden        INT,
    FOREIGN KEY(id_orden) REFERENCES ordenes(id_orden)
);

```

-- Tabla Detalle del carrito de compras

```

CREATE TABLE IF NOT EXISTS detalle_carrito
(
    id_detalle_carrito INT auto_increment PRIMARY KEY,
    cantidad            INT,
    precio_total        DECIMAL(10, 2) NOT NULL,
    id_carrito          INT,
    id_orden            INT,
    id_producto         INT,
    id_categoria        INT,
    id_proveedor        INT,
    FOREIGN KEY(id_carrito) REFERENCES carrito(id_carrito),
    FOREIGN KEY(id_orden) REFERENCES ordenes(id_orden),
    FOREIGN KEY(id_producto) REFERENCES productos(id_producto),
    FOREIGN KEY(id_categoria) REFERENCES categorias(id_categoria),
    FOREIGN KEY(id_proveedor) REFERENCES proveedores(id_proveedor)
);

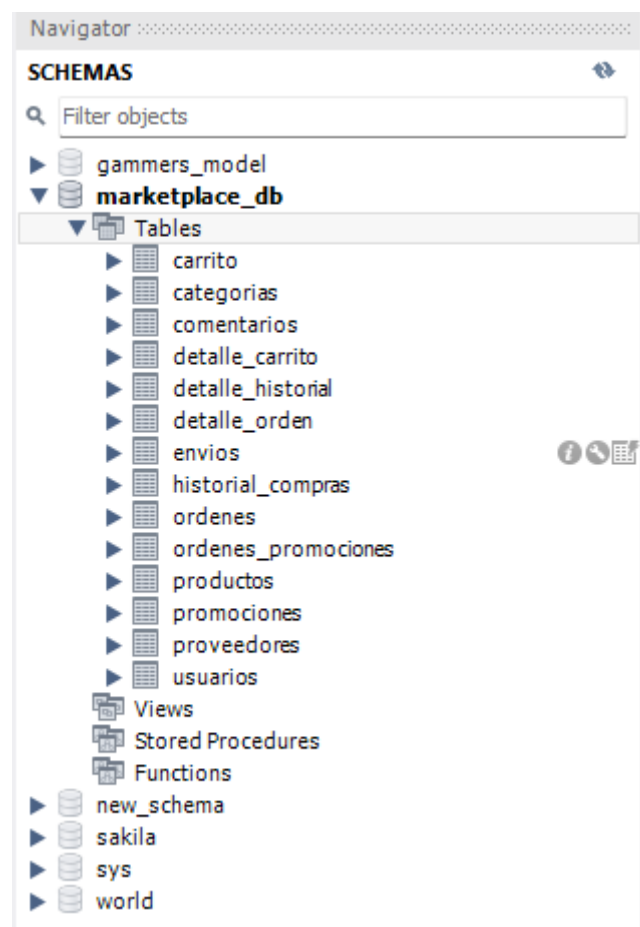
```

-- Tabla de Comentarios


```

CREATE TABLE IF NOT EXISTS comentarios
(
    id_comentario INT auto_increment PRIMARY KEY,
    comentario    VARCHAR(255) NOT NULL,
    fecha         DATETIME,
    estado        VARCHAR(45),
    calificacion  VARCHAR(45),
    id_producto   INT,
    id_categoria   INT,
    FOREIGN KEY(id_producto) REFERENCES productos(id_producto
),
    FOREIGN KEY(id_categoria) REFERENCES categorias(id_catego
ria)
);

```



Inserción de datos

```
INSERT INTO usuarios (id_usuario, tipo_documento, dni, nombre, apellido, correo, contraseña, telefono, fecha_nacimiento) VALUES
(1, 'DNI', 123456789, 'Juan', 'Pérez', 'juan.perez@email.com', 'contraseña123', '987654321', '1990-05-15 00:00:00'),
(2, 'Pasaporte', 987654321, 'Ana', 'Gómez', 'ana.gomez@email.com', 'clave_segura', '654321987', '1985-08-22 00:00:00'),
(3, 'DNI', 567890123, 'Pedro', 'Martínez', 'pedro.martinez@email.com', 'password123', '789012345', '1992-02-10 00:00:00'),
(4, 'Carné de Identidad', 456789012, 'Laura', 'García', 'laura.garcia@email.com', 'seguridad456', '012345678', '1988-11-30 00:00:00'),
(5, 'DNI', 345678901, 'Miguel', 'López', 'miguel.lopez@email.com', 'contraseña456', '345678901', '1995-07-05 00:00:00'),
(6, 'Pasaporte', 234567890, 'Carolina', 'Sánchez', 'carolina.sanchez@email.com', 'clave_segura789', '567890123', '1998-04-18 00:00:00'),
(7, 'DNI', 876543210, 'Daniel', 'Fernández', 'daniel.fernandez@email.com', 'secure_pass', '678901234', '1983-09-12 00:00:00'),
(8, 'Carné de Identidad', 765432109, 'Isabel', 'Díaz', 'isabel.diaz@email.com', 'password789', '890123456', '1991-12-25 00:00:00'),
(9, 'DNI', 654321098, 'Lucas', 'Hernández', 'lucas.hernandez@email.com', 'contraseña789', '901234567', '1980-06-08 00:00:00'),
(10, 'Pasaporte', 543210987, 'María', 'Jiménez', 'maria.jimenez@email.com', 'seguro_pass', '234567890', '1987-03-03 00:00:00');
```

```
INSERT INTO historial_compras (id_historial, fecha, estado, id_usuario)
VALUES
(1, '2024-06-01 10:15:00', 'Completado', 1),
(2, '2024-06-02 12:30:00', 'Pendiente', 2),
(3, '2024-06-03 14:45:00', 'Cancelado', 3),
(4, '2024-06-04 09:00:00', 'Completado', 4),
(5, '2024-06-05 11:20:00', 'Pendiente', 5),
(6, '2024-06-06 16:00:00', 'Completado', 6),
(7, '2024-06-07 08:45:00', 'Cancelado', 7),
(8, '2024-06-08 13:10:00', 'Completado', 8),
(9, '2024-06-09 15:30:00', 'Pendiente', 9),
(10, '2024-06-10 10:00:00', 'Completado', 10);
```

```
INSERT INTO categorias (id_categoria, nombre, descripcion, fecha_creacion )
VALUES
(1, 'Electrónicos', 'Productos electrónicos como teléfonos, computadoras y accesorios', NOW()),
(2, 'Ropa', 'Ropa de moda para hombres, mujeres y niños', NOW()),
(3, 'Hogar y Jardín', 'Artículos para el hogar, muebles y suministros de jardinería.', NOW()),
(4, 'Deportes', 'Equipos y accesorios para deportes y actividades al aire libre.', NOW()),
(5, 'Libros', 'Libros de diversas categorías', NOW()),
(6, 'Juguetes', 'Juguetes para niños de todas las edades', NOW()),
(7, 'Joyería', 'Joyas y accesorios', NOW()),
(8, 'Alimentación', 'Productos alimenticios y gourmet', NOW());
```

```
(9, 'Salud y Belleza', 'Productos para el cuidado personal', NOW()),
(10, 'Automotriz', 'Accesorios y productos para automóviles', NOW());
```

```
INSERT INTO productos (id_producto, nombre, descripcion, precio, cantidad)
VALUES
(1, 'Smartphone X', 'Potente smartphone con cámara dual', 599.99, 50),
(2, 'Camiseta de Algodón', 'Camiseta cómoda y de alta calidad', 19.99, 100),
(3, 'Juego de Sartenes', 'Set de sartenes antiadherentes', 89.95, 30),
(4, 'Balón de Fútbol', 'Balón oficial para partidos', 29.99, 20),
(5, 'Libro "Aventuras en el Espacio"', 'Novela de ciencia ficción', 14.50, 80),
(6, 'Muñeca Interactiva', 'Muñeca que habla y realiza acciones', 39.99, 15),
(7, 'Collar de Diamantes', 'Collar elegante con diamantes', 499.99, 10),
(8, 'Pack de Snacks Saludables', 'Variado pack de snacks saludables', 29.95, 50),
(9, 'Kit de Cuidado Facial', 'Productos para el cuidado facial', 59.99, 25),
(10, 'Limpiaparabrisas Universal', 'Limpiaparabrisas de calidad para automóviles', 12.75, 40);
```

```
INSERT INTO detalle_historial (precio, cantidad, comentario, id_historial, id_usuario, id_producto, id_categoria) VALUES
(59.99, 2, 'Buena compra', 1, 3, 1, 1),
(19.99, 1, 'Talla correcta', 1, 2, 2, 2),
(89.95, 1, 'Excelente calidad', 1, 1, 3, 3),
(29.99, 3, 'Divertido para jugar en equipo', 2, 4, 4, 4),
(14.50, 1, 'Interesante trama', 2, 3, 5, 5),
(39.99, 1, 'Ideal para regalar', 3, 1, 6, 6),
(499.99, 1, 'Muy elegante', 3, 2, 7, 7),
(29.95, 2, 'Deliciosos snacks', 4, 4, 8, 8),
(59.99, 1, 'Noté mejoras en mi piel', 4, 3, 9, 9),
(12.75, 1, 'Fácil instalación', 5, 1, 10, 10);
```

```
INSERT INTO proveedores (id_proveedor, nombre, contacto, direccion, estado, info_financiera, tipo) VALUES
(1, 'Proveedor A', 999888777, 'Calle Principal 123', 'Activo', 'Cuenta Bancaria: XXXX-XXXX-XXXX-XXXX', 'Electrónicos'),
(2, 'Proveedor B', 888777666, 'Avenida Secundaria 456', 'Activo', 'Cuenta Bancaria: YYYY-YYYY-YYYY-YYYY', 'Ropa'),
(3, 'Proveedor C', 777666555, 'Plaza Comercial 789', 'Inactivo', 'Cuenta Bancaria: ZZZZ-ZZZZ-ZZZZ-ZZZZ', 'Hogar'),
(4, 'Proveedor D', 666555444, 'Calle Peatonal 101', 'Activo', 'Cuenta Bancaria: WWWW-WWWW-WWWW-WWWW', 'Deportes'),
(5, 'Proveedor E', 555444333, 'Calle de los Libros 202', 'Inactivo', 'Cuenta Bancaria: VVVV-VVVV-VVVV-VVVV', 'Libros'),
(6, 'Proveedor F', 444333222, 'Esquina de los Juguetes 303', 'Activo', 'Cuenta Bancaria: UUUU-UUUU-UUUU-UUUU', 'Juguetes'),
(7, 'Proveedor G', 333222111, 'Avenida de las Joyas 404', 'Activo', 'Cuenta Bancaria: TTTT-TTTT-TTTT-TTTT', 'Joyería'),
(8, 'Proveedor H', 222111000, 'Calle de los Alimentos 505', 'Inactivo', 'Cuenta Bancaria: SSSS-SSSS-SSSS-SSSS', 'Alimentos');
```

```
, 'Cuenta Bancaria: SSSS-SSSS-SSSS-SSSS', 'Alimentación'),
(9, 'Proveedor I', 111000999, 'Paseo de la Salud 606', 'Activo', 'Cuenta Bancaria: RRRR-RRRR-RRRR-RRRR', 'Salud y Belleza'),
(10, 'Proveedor J', 999000888, 'Bulevar Automotriz 707', 'Activo', 'Cuenta Bancaria: QQQQ-QQQQ-QQQQ-QQQQ', 'Automotriz');
```

```
INSERT INTO promociones (id_promocion, nombre, descuento, fecha_inicio, fecha_fin, id_producto, id_categoria, id_proveedor) VALUES
(1, 'Oferta Verano', 0.15, '2024-06-01 00:00:00', '2024-06-30 23:59:59', 1, 1, 1),
(2, 'Descuento Ropa', 0.20, '2024-07-01 00:00:00', '2024-07-15 23:59:59', 2, 2, 2),
(3, 'Liquidación Hogar', 0.30, '2024-08-01 00:00:00', '2024-08-31 23:59:59', 3, 3, 3),
(4, 'Especial Deportes', 0.25, '2024-09-01 00:00:00', '2024-09-15 23:59:59', 4, 4, 4),
(5, 'Promo Libros', 0.18, '2024-10-01 00:00:00', '2024-10-31 23:59:59', 5, 5, 5),
(6, 'Juguetes Rebajados', 0.22, '2024-11-01 00:00:00', '2024-11-15 23:59:59', 6, 6, 6),
(7, 'Descuento Joyería', 0.15, '2024-12-01 00:00:00', '2024-12-31 23:59:59', 7, 7, 7),
(8, 'Oferta Alimentación', 0.25, '2025-01-01 00:00:00', '2025-01-15 23:59:59', 8, 8, 8),
(9, 'Cuidado Facial en Oferta', 0.20, '2025-02-01 00:00:00', '2025-02-28 23:59:59', 9, 9, 9),
(10, 'Accesorios Automotrices', 0.15, '2025-03-01 00:00:00', '2025-03-15 23:59:59', 10, 10, 10);
```

```
INSERT INTO ordenes (id_orden, fecha, estado, id_promocion) VALUES
(1, '2024-06-01 10:15:00', 'En Proceso', 1),
(2, '2024-06-02 12:30:00', 'Enviado', 2),
(3, '2024-06-03 14:45:00', 'Entregado', 3),
(4, '2024-06-04 09:00:00', 'En Proceso', 4),
(5, '2024-06-05 11:20:00', 'Enviado', 5),
(6, '2024-06-06 16:00:00', 'Entregado', 6),
(7, '2024-06-07 08:45:00', 'En Proceso', 7),
(8, '2024-06-08 13:10:00', 'Enviado', 8),
(9, '2024-06-09 15:30:00', 'Entregado', 9),
(10, '2024-06-10 10:00:00', 'Enviado', 10);
```

```
INSERT INTO envios (id_envio, direccion, estado, fecha_entrega, metodo, courier, costo, tiempo_entrega) VALUES
(1, 'Calle Principal 123', 'En Proceso', '2024-06-15 14:00:00', 'Estándar', 'EnvíoExpress', 15.99, NULL),
(2, 'Avenida Secundaria 456', 'Enviado', '2024-06-16 10:30:00', 'Express', 'RápidoEnvíos', 25.50, '2024-06-18 14:00:00'),
(3, 'Plaza Comercial 789', 'Entregado', '2024-06-17 11:45:00', 'Estándar', 'MegaLogística', 12.75, '2024-06-19 12:30:00'),
(4, 'Calle Peatonal 101', 'En Proceso', '2024-06-18 09:30:00', 'Express', 'EntregaVeloz', 20.00, NULL),
(5, 'Calle de los Libros 202', 'Enviado', '2024-06-
```

```

19 12:15:00', 'Estándar', 'LibrosExpress', 18.99, '2024-06-
21 15:00:00'),
(6, 'Esquina de los Juguetes 303', 'Entregado', '2024-06-
20 15:30:00', 'Express', 'JugueteRápido', 22.50, '2024-06-
22 10:45:00'),
(7, 'Avenida de las Joyas 404', 'Enviado', '2024-06-
21 14:45:00', 'Estándar', 'JoyasEnvíos', 30.00, '2024-06-
23 13:00:00'),
(8, 'Calle de los Alimentos 505', 'En Proceso', '2024-06-
22 10:00:00', 'Express', 'RicoEnvío', 15.75, NULL),
(9, 'Paseo de la Salud 606', 'Entregado', '2024-06-
23 12:30:00', 'Estándar', 'SaludableExpress', 28.99, '2024-06-
25 09:15:00'),
(10, 'Bulevar Automotriz 707', 'En Proceso', '2024-06-
24 09:45:00', 'Express', 'AutoRápido', 19.50, NULL);

```

```

INSERT INTO detalle_orden (id_detalle, cantidad, precio_total, id_producto, id_categoria, id_usuario, id_envio, id_orden) VALUES
(1, 2, 35.99, 1, 1, 1, 1, 1),
(2, 1, 25.50, 2, 2, 2, 2, 2),
(3, 3, 45.75, 3, 3, 3, 3, 3),
(4, 4, 80.00, 4, 4, 4, 4, 4),
(5, 2, 37.98, 5, 5, 5, 5, 5),
(6, 1, 44.99, 6, 6, 6, 6, 6),
(7, 3, 90.00, 7, 7, 7, 7, 7),
(8, 2, 31.50, 8, 8, 8, 8, 8),
(9, 1, 28.99, 9, 9, 9, 9, 9),
(10, 4, 65.00, 10, 10, 10, 10, 10);

```

```

INSERT INTO carrito (id_carrito, fecha_creacion, id_orden) VALUES
(1, '2024-06-01 10:15:00', 1),
(2, '2024-06-02 12:30:00', 2),
(3, '2024-06-03 14:45:00', 3),
(4, '2024-06-04 09:00:00', 4),
(5, '2024-06-05 11:20:00', 5),
(6, '2024-06-06 16:00:00', 6),
(7, '2024-06-07 08:45:00', 7),
(8, '2024-06-08 13:10:00', 8),
(9, '2024-06-09 15:30:00', 9),
(10, '2024-06-10 10:00:00', 10);

```

```

INSERT INTO detalle_carrito (id_detalle_carrito, cantidad, precio_total, id_carrito, id_orden, id_producto, id_categoria, id_proveedor) VALUES
(1, 2, 35.99, 1, 1, 1, 1, 4),
(2, 1, 25.50, 2, 2, 2, 2, 4),
(3, 3, 45.75, 3, 3, 3, 3, 4),
(4, 4, 80.00, 4, 4, 4, 4, 4),
(5, 2, 37.98, 5, 5, 5, 5, 5),
(6, 1, 44.99, 6, 6, 6, 6, 6),
(7, 3, 90.00, 7, 7, 7, 7, 7),
(8, 2, 31.50, 8, 8, 8, 8, 8),
(9, 1, 28.99, 9, 9, 9, 9, 9),
(10, 4, 65.00, 10, 10, 10, 10, 10);

```

```
INSERT INTO comentarios (id_comentario, comentario, fecha, estado, c
alificacion, id_producto, id_categoria) VALUES
(1, '¡Excelente producto, estoy muy satisfecho!', '2024-06-
01 14:30:00', 'Aprobado', '5 estrellas', 1, 1),
(2, 'Buen servicio al cliente, rápido y eficiente.', '2024-06-
02 11:45:00', 'Aprobado', '4 estrellas', 2, 2),
(3, 'Producto de calidad, lo recomiendo.', '2024-06-
03 09:20:00', 'Pendiente', '5 estrellas', 3, 3),
(4, 'No recibí mi pedido a tiempo, servicio insatisfactorio.', '2024
-06-04 16:00:00', 'Rechazado', '2 estrellas', 4, 4),
(5, '¡Me encanta! Compré otro para regalar.', '2024-06-
05 12:15:00', 'Aprobado', '5 estrellas', 5, 5),
(6, 'El producto no cumplió con mis expectativas.', '2024-06-
06 14:45:00', 'Aprobado', '3 estrellas', 6, 6),
(7, 'Servicio de entrega excelente, llegó antes de lo esperado.', '2
024-06-07 10:30:00', 'Aprobado', '4 estrellas', 7, 7),
(8, 'Calidad regular, esperaba más.', '2024-06-
08 13:00:00', 'Rechazado', '2 estrellas', 8, 8),
(9, '¡Me encanta este producto! Compraría de nuevo.', '2024-06-
09 15:30:00', 'Pendiente', '5 estrellas', 9, 9),
(10, 'Buen servicio, producto conforme a la descripción.', '2024-06-
10 08:45:00', 'Aprobado', '4 estrellas', 10, 10);
```

Vistas Principales

1. Historial de compras por cada Usuario

```
CREATE OR REPLACE VIEW vista_historial_usuario AS

SELECT u.id_usuario, u.tipo_documento, u.numero_documento, u.nombre,
u.apellido, u.correo, u.telefono, u.fecha_nacimiento, h.id_historial, h.fecha, h.estado

FROM usuarios u

JOIN historial_compras h ON u.id_usuario = h.id_usuario;

SELECT * FROM vista_historial_usuario;
```

Result Grid											
Filter Rows:											
Export: Wrap Cell Contents:											
	id_usuario	tipo_documento	nroDoc	nombre	apellido	correo	telefono	fecha_nacimiento	id_historial	fecha	estado
▶	1	DNI	123456789	Juan	Pérez	juan.perez@email.com	987654321	1990-05-15 00:00:00	1	2024-06-01 10:15:00	Completado
1		DNI	123456789	Juan	Pérez	juan.perez@email.com	987654321	1990-05-15 00:00:00	11	2024-06-08 13:10:00	Cancelado
2		Pasaporte	987654321	Ana	Gómez	ana.gomez@email.com	654321987	1985-08-22 00:00:00	2	2024-06-02 12:30:00	Pendiente
2		Pasaporte	987654321	Ana	Gómez	ana.gomez@email.com	654321987	1985-08-22 00:00:00	12	2024-06-09 15:30:00	Completado
3		DNI	567890123	Pedro	Martínez	pedro.martinez@email.com	789012345	1992-02-10 00:00:00	3	2024-06-03 14:45:00	Cancelado
3		DNI	567890123	Pedro	Martínez	pedro.martinez@email.com	789012345	1992-02-10 00:00:00	13	2024-06-10 10:00:00	Pendiente
4		Carné de Identidad	456789012	Laura	García	laura.garcia@email.com	012345678	1988-11-30 00:00:00	4	2024-06-04 09:00:00	Completado
5		DNI	345678901	Miguel	López	miguel.lopez@email.com	345678901	1995-07-05 00:00:00	5	2024-06-05 11:20:00	Pendiente
6		Pasaporte	234567890	Carolina	Sánchez	carolina.sanchez@email.com	567890123	1998-04-18 00:00:00	6	2024-06-06 16:00:00	Completado
7		DNI	876543210	Daniel	Fernández	daniel.fernandez@email.com	678901234	1983-09-12 00:00:00	7	2024-06-07 08:45:00	Cancelado

2. Detalle historial de compras

```
CREATE OR REPLACE VIEW vista_carrito_detalles AS

SELECT dc.id_detalle_carrito, dc.cantidad, dc.precio_unitario,

       c.id_carrito, c.fecha_creacion,

       p.id_producto, p.nombre AS nombre_producto, p.description,

       pr.id_proveedor, pr.nombre AS nombre_proveedor,

       COUNT(*) AS total_detalles,

       SUM(dc.precio_unitario * dc.cantidad) AS total_precio

FROM detalle_carrito dc

JOIN carrito c ON dc.id_carrito = c.id_carrito

JOIN productos p ON dc.id_producto = p.id_producto

JOIN proveedores pr ON dc.id_proveedor = pr.id_proveedor

GROUP BY dc.id_detalle_carrito, c.id_carrito, p.id_producto

ORDER BY c.fecha_creacion;

SELECT * FROM vista_carrito_detalles;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Contents: <input checked="" type="checkbox"/>												
	id_detalle_carrito	cantidad	precio_unitario	id_carrito	fecha_creacion	id_producto	nombre_producto	description	id_proveedor	nombre_proveedor	total_detalles	total_precio
▶	1	2	35.99	1	2024-06-01 10:15:00	1	Smartphone X	Potente smartphone con cámara dual	1	Proveedor A	1	71.98
	2	1	25.50	2	2024-06-02 12:30:00	2	Camiseta de Algodón	Camiseta cómoda y de alta calidad	2	Proveedor B	1	25.50
	3	3	45.75	3	2024-06-03 14:45:00	3	Juego de Sartenes	Set de sartenes antiadherentes	3	Proveedor C	1	137.25
	4	4	80.00	4	2024-06-04 09:00:00	4	Balón de Fútbol	Balón oficial para partidos	4	Proveedor D	1	320.00
	5	2	37.98	5	2024-06-05 11:20:00	5	Libro "Aventuras en el Espacio"	Novela de ciencia ficción	5	Proveedor E	1	75.96
	6	1	44.99	6	2024-06-06 16:00:00	6	Muñeca Interactiva	Muñeca que habla y realiza acciones	6	Proveedor F	1	44.99
	7	3	90.00	7	2024-06-07 08:45:00	7	Collar de Diamantes	Collar elegante con diamantes	7	Proveedor G	1	270.00
	8	2	31.50	8	2024-06-08 13:10:00	8	Pack de Snacks Saludables	Variado pack de snacks saludables	8	Proveedor H	1	63.00
	9	1	28.99	9	2024-06-09 15:30:00	9	Kit de Cuidado Facial	Productos para el cuidado facial	9	Proveedor I	1	28.99
	10	4	65.00	10	2024-06-10 10:00:00	10	Limpiaparabrisas Universal	Limpiaparabrisas de calidad para automóviles	10	Proveedor J	1	260.00

3. Total Ventas por Categoría

```
CREATE OR REPLACE VIEW vista_total_venta_por_categoria AS

SELECT c.id_categoria, c.nombre AS nombre_categoria,

       COUNT(dh.id_detalle_historial) AS total_ventas,

       SUM(dh.precio * dh.cantidad) AS ingreso_total

FROM categorias c

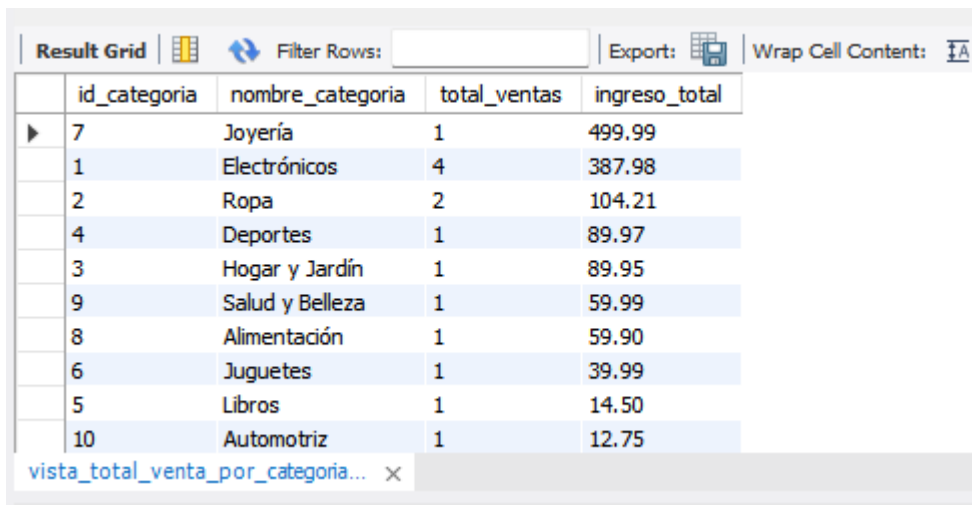
LEFT JOIN productos p ON c.id_categoria = p.id_categoria

LEFT JOIN detalle_historial dh ON p.id_producto = dh.id_producto

GROUP BY c.id_categoria, c.nombre

ORDER BY ingreso_total DESC;

SELECT * FROM vista_total_venta_por_categoria;
```



The screenshot shows a database application interface with a 'Result Grid' tab. The grid displays the results of a query on the view 'vista_total_venta_por_categoria'. The columns are 'id_categoria', 'nombre_categoria', 'total_ventas', and 'ingreso_total'. The data is sorted by 'ingreso_total' in descending order. The interface includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox.




	id_categoria	nombre_categoria	total_ventas	ingreso_total
▶	7	Joyería	1	499.99
	1	Electrónicos	4	387.98
	2	Ropa	2	104.21
	4	Deportes	1	89.97
	3	Hogar y Jardín	1	89.95
	9	Salud y Belleza	1	59.99
	8	Alimentación	1	59.90
	6	Juguetes	1	39.99
	5	Libros	1	14.50
	10	Automotriz	1	12.75

vista_total_venta_por_categoria... x

4. Total Ventas por Categoría Ordenes en estado “En Proceso”

```
CREATE OR replace VIEW vista_ordenes_pendientes
AS
    SELECT o.id_orden,
           o.fecha          AS order_fecha,
           o.estado         AS order_estado,
           c.fecha_creacion AS cart_fecha_creacion,
           u.nombre         AS user_nombre,
           u.apellido       AS user_apellido
    FROM   ordenes o
           join carrito c
             ON o.id_orden = c.id_orden
           join usuarios u
             ON c.id_orden = u.id_usuario
    WHERE  o.estado = 'En Proceso';

SELECT * FROM vista_ordenes_pendientes ;
```

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 						
	id_orden	order_fecha	order_estado	cart_fecha_creacion	user_nombre	user_apellido
▶	1	2024-06-01 10:15:00	En Proceso	2024-06-01 10:15:00	Juan	Pérez
	4	2024-06-04 09:00:00	En Proceso	2024-06-04 09:00:00	Laura	García
	7	2024-06-07 08:45:00	En Proceso	2024-06-07 08:45:00	Daniel	Fernández

5. Productos que tienen promociones

```
CREATE OR replace VIEW vista_productos_con_promociones AS
SELECT pr.nombre AS producto_nombre, pr.precio AS producto_pre
cio, c.nombre AS category_nombre, prom.nombre
AS promocion_nombre, prom.descuento AS promocion_descuento
FROM   productos pr
left join categorias c
ON pr.id_categoria = c.id_categoria
left join promociones prom
ON pr.id_producto = prom.id_producto
OR pr.id_categoria = prom.id_categoria;

SELECT * FROM vista_productos_con_promociones;
```

	producto_nombre	producto_precio	category_nombre	promocion_nombre	promocion_descuento
▶	Smartphone X	599.99	Electrónicos	Oferta Verano	0.15
	Camiseta de Algodón	19.99	Ropa	Oferta Verano Divertikids	0.35
	Camiseta de Algodón	19.99	Ropa	Descuento Ropa	0.20
	Juego de Sartenes	89.95	Hogar y Jardín	Liquidación Hogar	0.30
	Balón de Fútbol	29.99	Deportes	Especial Deportes	0.25
	Libro "Aventuras en el Espacio"	14.50	Libros	Oferta Verano EnjoyFamily	0.55
	Libro "Aventuras en el Espacio"	14.50	Libros	Promo Libros	0.18
	Muñeca Interactiva	39.99	Juguetes	Juguetes Rebajados	0.22
	Collar de Diamantes	499.99	Joyería	Descuento Joyería	0.15
	Pack de Snacks Saludables	29.95	Alimentación	Oferta Alimentación	0.25

Funciones

1. Historial de compras por Usuario

```
USE marketplace_db;

DELIMITER //
CREATE FUNCTION obtener_historial_compras_por_usuario
(usuario_id int)
RETURNS varchar(255)
DETERMINISTIC
BEGIN
DECLARE resultado VARCHAR(255);
SELECT Group_concat(Concat('ID Historial: ', hc.id_historial,
', Fecha: ', hc.fecha, ', Estado: ', hc.estado, ', Productos: ', p.productos))
INTO resultado
FROM historial_compras hc
JOIN
(
        SELECT dh.id_historial,
        Group_concat(p.nombre) AS productos
        FROM detalle_historial dh
        JOIN productos p
        ON dh.id_producto = p.id_producto
        GROUP BY dh.id_historial ) p
ON hc.id_historial = p.id_historial
WHERE hc.id_usuario = usuario_id;
RETURN resultado;

END //
DELIMITER ;

SELECT obtener_historial_compras_por_usuario (1);
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
obtener_historial_compras_por_usuario(1)				
ID Historial: 1, Fecha: 2024-06-01 10:15:00, Estado: Completado, Productos: Smartphone X,Smartphone X,Smartphone X,Camiseta de Algodón,Camiseta de Algodón,Juego de Sartenes				

Result 4 x

2. Productos por categoría específica

```
DELIMITER //
CREATE FUNCTION obtener_productos_por_categoria
(categoria_nombre varchar(45))
RETURNS varchar(255)
DETERMINISTIC
BEGIN
DECLARE resultado VARCHAR(255);
SELECT Group_concat(p.nombre)
  INTO resultado
  FROM productos p
  JOIN categorias c
    ON p.id_categoria = c.id_categoria
  WHERE c.nombre = categoria_nombre;
RETURN resultado;

END //
DELIMITER ;
```

```
SELECT obtener_productos_por_categoria
('Electrónicos') AS Resultado;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Resultado			
Smartphone X, Smartwatch			

3. Calcular Descuento Total promoción por Ordenes

```
DELIMITER //
CREATE FUNCTION calcular_descuento_total(orden_id int)
RETURNS decimal(10, 2)
DETERMINISTIC
BEGIN
DECLARE descuento_total DECIMAL(10, 2);
SELECT Sum(p.descuento)
  INTO descuento_total
  FROM promociones p
  JOIN ordenes_promociones op
    ON p.id_promocion = op.id_promocion
  JOIN ordenes o
    ON op.id_orden = o.id_orden
  WHERE o.id_orden = orden_id;
RETURN descuento_total;

END //
DELIMITER ;
```

```
SELECT calcular_descuento_total(1);
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
calcular_descuento_total(1)			
1.05			

Stored Procedures

1. Calcular el Monto total por Orden

```
USE marketplace_db;

DELIMITER //

CREATE PROCEDURE calcular_total_orden(IN orden_id INT )
BEGIN
    DECLARE total DECIMAL(10, 2);

    SELECT Sum(do.precio_unitario*do.cantidad)
        INTO total
        FROM detalle_orden DO
        WHERE do.id_orden = orden_id;

    SELECT total AS TotalOrden;

END //
DELIMITER ;

CALL calcular_total_orden(1);
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
TotalOrden				
▶	243.96			

2. Orden por Nombre de la tabla, Columna y Orden

```
DELIMITER //

CREATE PROCEDURE ordenar_tabla
    (IN p_nombre_tabla varchar(255),
    IN p_campo_ordenamiento varchar(255),
    IN p_orden varchar(4) )
BEGIN
    SET @sql_query = Concat('SELECT * FROM ',
    p_nombre_tabla, ' ORDER BY ',
    p_campo_ordenamiento, ' ', p_orden);

    PREPARE stmt FROM @sql_query;

    EXECUTE stmt;

    DEALLOCATE prepare stmt;
```

```

END //
DELIMITER ;

CALL ordenar_tabla('comentarios','estado','DESC');

```

Result Grid Filter Rows: Export: Wrap Cell Content:							
	id_comentario	comentario	fecha	estado	calificacion	id_producto	id_categoria
▶	4	No recibí mi pedido a tiempo, servicio insatisfact...	2024-06-04 16:00:00	Rechazado	2 estrellas	4	4
	8	Calidad regular, esperaba más.	2024-06-08 13:00:00	Rechazado	2 estrellas	8	8
	3	Producto de calidad, lo recomiendo.	2024-06-03 09:20:00	Pendiente	5 estrellas	3	3
	9	¡Me encanta este producto! Compraría de nuevo.	2024-06-09 15:30:00	Pendiente	5 estrellas	9	9
	1	¡Excelente producto, estoy muy satisfecho!	2024-06-01 14:30:00	Aprobado	5 estrellas	1	1
	2	Buen servicio al cliente, rápido y eficiente.	2024-06-02 11:45:00	Aprobado	4 estrellas	2	2
	5	¡Me encanta! Compré otro para regalar.	2024-06-05 12:15:00	Aprobado	5 estrellas	5	5
	6	El producto no cumplió con mis expectativas.	2024-06-06 14:45:00	Aprobado	3 estrellas	6	6
	7	Servicio de entrega excelente, llegó antes de lo...	2024-06-07 10:30:00	Aprobado	4 estrellas	7	7

3. Eliminar un ítem del carrito de compras

```

DELIMITER //
CREATE PROCEDURE eliminar_item_carrito
(IN carrito_id INT, IN detalle_carrito_id INT)
BEGIN
    DECLARE existe_carrito INT;
    SELECT Count(*) INTO existe_carrito FROM carrito
    WHERE id_carrito = carrito_id;

    -
    - Si el carrito existe, eliminar el detalle del carrito
    IF existe_carrito > 0 then
        DELETE
        FROM detalle_carrito
        WHERE id_detalle_carrito = detalle_carrito_id;
    SELECT 'Item eliminado del carrito correctamente.' AS mensaj
    e ; ELSE
    SELECT 'El carrito especificado no existe.' AS mensaje;
    ENDIF;
END //
DELIMITER;

CALL eliminar_item_carrito(10, 10);

SELECT * FROM detalle_carrito;

```

Result Grid Filter Rows: Export: Wrap Cell Content:	
	mensaje
▶	Item eliminado del carrito correctamente.

4. Crear un ítem del carrito de compras

```
DELIMITER //
```

```
CREATE PROCEDURE agregar_item_carrito( IN carrito_id INT ,  
IN producto_id INT ,IN cantidad INT ,  
IN precio_unitario decimal(10, 2) )  
BEGIN  
  DECLARE existe_carrito INT ;  
  -- Verificar si el carrito existe  
  SELECT Count(*) INTO existe_carrito FROM carrito  
  WHERE id_carrito = carrito_id;  
  
  -- Si el carrito existe, agregar el detalle al carrito  
  IF existe_carrito > 0 then  
    INSERT INTO detalle_carrito  
      (  
        id_carrito,  
        id_producto,  
        cantidad,  
        precio_unitario  
      )  
    VALUES  
      (  
        carrito_id,  
        producto_id,  
        cantidad,  
        precio_unitario  
      );  
  SELECT 'Item agregado al carrito correctamente.' AS mensaje;  
  ELSE SELECT 'El carrito especificado no existe.' AS mensaje;  
  ENDIF;  
END //
```

```
DELIMITER ;
```

```
CALL agregar_item_carrito(13, 8, 3, 29.95);
```

```
SELECT * FROM detalle_carrito;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	mensaje			
▶	Item agregado al carrito correctamente.			


Triggers

Creación de la Tabla log de productos

```
USE marketplace_db;

CREATE TABLE IF NOT EXISTS log_productos
(
    log_id int auto_increment PRIMARY KEY,
    id_producto int NOT NULL,
    usuario_modificacion VARCHAR(50),
    fecha_modificacion date NOT NULL,
    hora_modificacion time NOT NULL,
    accion_realizada varchar(255) NOT NULL,
    columna_modificada varchar(255) NULL,
    valor_anterior varchar(255) NULL,
    valor_actual varchar(255) NULL
);

SELECT * FROM log_productos;
```



	log_id	id_producto	usuario_modificacion	fecha_modificacion	hora_modificacion	accion_realizada	columna_modificada	valor_anterior	valor_actual
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trigger 1 : Creación de un Trigger BEFORE UPDATE

```
DELIMITER //
```

```
CREATE TRIGGER before_update_log_productos
```

```
BEFORE UPDATE on productos
```

```
FOR EACH ROW
```

```
BEGIN
```



```
INSERT INTO log_productos (id_producto, usuario_modificacion,
fecha_modificacion, hora_modificacion, accion_realizada,
valor_anterior, valor_actual)
```

```
VALUES (NEW.id_producto, 'nombre_de_usuario', CURDATE(),
CURTIME(), 'before update', NEW.nombre, NEW.nombre);
```

```
END //
DELIMITER ;
```

```
UPDATE productos

SET precio =23.51

WHERE id_producto=2
```

```
SELECT * FROM productos;

SELECT * FROM log_productos;
```

log_id	id_producto	usuario_modificacion	fecha_modificacion	hora_modificacion	accion_realizada	columna_modificada	valor_anterior	valor_actual
1	2	nombre_de_usuario	2024-01-27	13:24:11	update	NULL	Camiseta de Algodón	Camiseta de Algodón
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trigger 2: Creación Trigger AFTER UPDATE

```
DELIMITER //

CREATE TRIGGER after_update_producto_log_trigger
AFTER UPDATE ON productos
FOR EACH ROW
BEGIN
    DECLARE columna_modificada VARCHAR(255);
    DECLARE valor_anterior VARCHAR(255);
    DECLARE valor_actual VARCHAR(255);

    -- Determinar la columna modificada
    IF old.precio <> new.precio then
        SET columna_modificada = precio;

        SET valor_anterior = old.precio;
        SET valor_actual = new.precio;
```

```

ELSEIF old.cantidad <> new.cantidad THEN
SET columna_modificada = 'cantidad';

SET valor_anterior = old.cantidad;

SET valor_actual = new.cantidad;

ELSEIF old.descripcion <> new.Descripcion THEN
SET columna_modificada = 'descripcion';

SET valor_anterior = old.description;

SET valor_actual = new.descripcion;

ELSEIF old.nombre <> new.nombre
THEN SET columna_modificada = 'nombre';

SET valor_anterior = old.nombre;

SET valor_actual = new.nombre;

END IF;

-- Insertar en el log solo si alguna columna ha cambiado

IF columna_modificada IS NOT NULL THEN

INSERT INTO log_productos (id_producto, usuario_modificacion,
fecha_modificacion, hora_modificacion, accion_realizada,
columna_modificada, valor_anterior, valor_actual)

VALUES (NEW.id_producto, CURRENT_USER(), CURDATE(), CURTIME(),
'After update', columna_modificada, valor_anterior,
valor_actual);

END IF;

END //
DELIMITER ;

UPDATE productos

SET precio =23.55

WHERE id_producto=2;

SELECT * FROM productos;

SELECT * FROM log_productos;

```


Trigger 3: Crear trigger BEFORE UPDATE

```

DELIMITER //

CREATE TRIGGER before_update_log_usuarios
BEFORE UPDATE ON usuarios
FOR EACH ROW
BEGIN
IF NEW.tipo_documento <> OLD.tipo_documento THEN
INSERT INTO log_usuarios (id_usuario, campo_modificado,
valor_anterior, valor_actual, fecha_modificacion,
hora_modificacion)
VALUES (OLD.id_usuario, 'tipo_documento', OLD.tipo_documento,
NEW.tipo_documento, CURDATE(), CURTIME());
END IF;

IF NEW.numero_documento <> OLD.numero_documento THEN
INSERT INTO log_usuarios (id_usuario, campo_modificado,
valor_anterior, valor_actual, fecha_modificacion,
hora_modificacion)
VALUES (OLD.id_usuario, 'numero_documento ', OLD.
numero_documento, NEW.numero_documento, CURDATE(),
CURTIME());

END IF;

END;

//

DELIMITER ;

SELECT * FROM usuarios;

SELECT * FROM log_usuarios;

UPDATE usuarios
SET numero_documento ="123456333"
WHERE id_usuario=1

```

	log_id	id_usuario	campo_modificado	valor_anterior	valor_actual	fecha_modificacion	hora_modificacion
▶	1	1	nroDoc	123456789	123456333	2024-01-27	13:56:37
✱	NULL	NULL		NULL	NULL	NULL	NULL

Trigger 4: Crear trigger AFTER INSERT

```
DELIMITER //
```

```
CREATE TRIGGER after_insert_log_usuarios
AFTER INSERT ON usuarios

FOR EACH ROW

BEGIN

INSERT INTO log_usuarios (id_usuario, campo_modificado,
valor_anterior , valor_actual , fecha_modificacion,
hora_modificacion)

VALUES (NEW.id_usuario, 'nuevo_usuario_insertado', NULL, NULL,
CURDATE(), CURTIME());

END;

//

DELIMITER ;

INSERT INTO usuarios (id_usuario, tipo_documento,
numero_documento, nombre, apellido, correo, contrasena, telefono,
fecha_nacimiento) VALUES

(11, 'DNI', 123456111, 'José', 'Pérez', 'jose.perez@email.com',
'contraseña123', '987654321', '1990-05-15 00:00:00');

SELECT * FROM log usuarios;
```

[illegible]

Definición de usuarios y permisos

```
DROP user solo_lectura@floreria;
```

```
DROP user lectura_insercion_modificacion@floreria;
```

```
-- Usuario solo_lectura con permisos de solo lectura
```

```
CREATE USER solo_lectura@floreria IDENTIFIED BY '9273@]=maow';
```

```
- Conceder permisos de solo lectura en todas las tablas del marketplace_db. No se permite eliminar registros
```

```
GRANT SELECT ON marketplace_db.* TO solo_lectura@floreria;
```

```
SHOW grants FOR solo_lectura@floreria;
```

```
-
```

```
- Usuario lectura_insercion_modificacion con permisos de lectura, inserción y modificación
```

```
CREATE USER 'lectura_insercion_modificacion'@floreria IDENTIFIED BY 'j28jw157h#'
```

```
;
```

```
- Conceder permisos de lectura, inserción y modificación en todas las tablas del marketplace_db. No se permite eliminar registros
```

```
GRANT SELECT, INSERT, UPDATE ON marketplace_db.* TO 'lectura_insercion_modificacion'@floreria;
```

```
SHOW grants FOR lectura_insercion_modificacion@floreria;
```

```
-
```

```
- Eliminar la capacidad de eliminar registros para ambos usuarios y que se aplica a nivel global
```

```
REVOKE DELETE ON *.* FROM solo_lectura@floreria;
```

```
REVOKE DELETE ON *.* FROM lectura_insercion_modificacion@floreria;
```

Transacciones

```
USE marketplace_db;

SELECT * FROM proveedores;

-- Iniciar una transacción en la tabla 'productos'

START TRANSACTION;

-
- Eliminar algunos registros en la tabla 'product' (o reemplazar por
  inserción si está vacía)

DELETE FROM proveedores WHERE id_proveedor = 1;

DELETE FROM proveedores WHERE id_proveedor = 2;

DELETE FROM proveedores WHERE id_proveedor = 3;

-- Comentar la sentencia Rollback (no ejecutar)
-- ROLLBACK;

-- Comentar la sentencia Commit (no ejecutar)
-- COMMIT;

-
- INSERT INTO proveedores (id_proveedor,nombre, contacto, direccion,
  estado, info_financiera, tipo) VALUES
-
- (1,'Proveedor A', 999888777, 'Calle Principal 123', 'Activo', 'Cue
nta Bancaria: XXXX-XXXX-XXXX-XXXX', 'Electrónicos'),
-
- (2,'Proveedor B', 888777666, 'Avenida Secundaria 456', 'Activo', '
Cuenta Bancaria: YYYY-YYYY-YYYY-YYYY', 'Ropa'),
-
- (3,'Proveedor C', 777666555, 'Plaza Comercial 789', 'Inactivo', 'C
uenta Bancaria: ZZZZ-ZZZZ-ZZZZ-ZZZZ', 'Hogar');
-- Iniciar una transacción en la tabla 'review'
```

```

START TRANSACTION;
-- Insertar ocho nuevos registros en la tabla productos

INSERT INTO productos
    (
        id_producto,
        nombre,
        description,
        price,
        cantidad,
        id_categoria
    )
VALUES
    (
        14,
        'Auriculares Inalámbricos',
        'Auriculares Bluetooth con cancelación de ruido',
        79.99,
        50,
        1
    ),
    (
        15,
        'Bicicleta de Montaña',
        'Bicicleta todo terreno con suspensión',
        499.95,
        15,
        2
    ),
    (
        16,
        'Cámara DSLR Profesional',
        'Cámara de alta resolución para fotografía profesional',
        899.99,

```



```

        10,
        3
    )
,
(
    17,
    'Mochila Impermeable',
    'Mochila resistente al agua para actividades
al aire libre',
    45.50,
    30,
    4
);

-- Agregar un savepoint después de la inserción del cuarto registro
SAVEPOINT savepoint4;
-- Continuar con la inserción de registros

INSERT INTO productos
(
    id_producto,
    nombre,
    description,
    price,
    cantidad,
    id_categoria
)
VALUES
(
    18,
    'Impresora Multifunción',
    'Impresora que imprime, escanea y copia',
    129.75,
    20,
    5
)
,

```

```

        (
            19,
            'Teclado Gaming RGB',
            'Teclado mecánico para juegos con retroilumi
nación RGB',
            69.99,
            25,
            6
        )
    ,
    (
        20,
        'Silla Ergonómica de Oficina',
        'Silla ajustable para una postura cómoda dur
ante horas de trabajo',
        149.95,
        15,
        7
    )
    ,
    (
        21,
        'Estuche de Maquillaje Profesional',
        'Kit completo de maquillaje para uso profes
ional',
        79.99,
        20,
        8
    )
);

```

-- Agregar un savepoint después de la inserción del octavo registro

```
SAVEPOINT savepoint8;
```

- Comentar la sentencia para eliminar el savepoint de los primeros 4 registros insertados

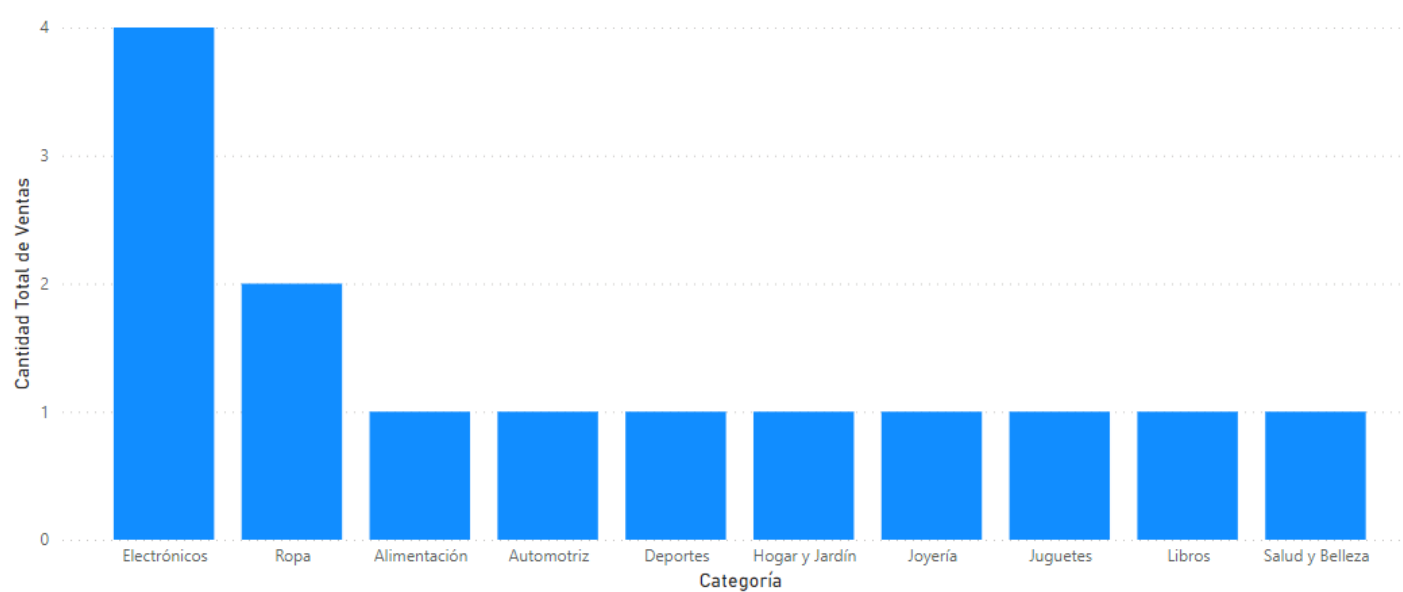
-- ROLLBACK TO savepoint4;

```
SELECT * FROM productos;
```

Reportes

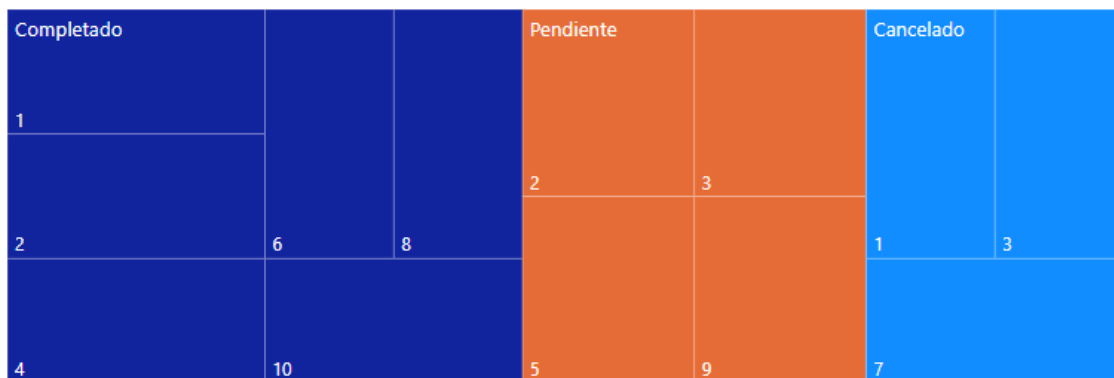
En el siguiente gráfico, se presenta la cantidad total de ventas desglosada por categoría. Esta representación visual facilita una comprensión más sencilla de los productos que gozan de mayor preferencia entre nuestros clientes. Este análisis nos permite identificar con claridad los productos más populares, proporcionándonos información valiosa para desarrollar estrategias de promoción efectivas y mejorar continuamente la calidad de dichos productos.

Cantidad Total de Ventas por Categoría



En el gráfico a continuación, se exhibe el historial de compras de cada usuario en función del estado de la orden. Esta representación visual simplifica la comprensión de las órdenes que se encuentran en estado completado, pendiente y cancelado. Este análisis proporciona una visión clara de por qué un cliente no finalizó una compra, permitiéndonos enfocarnos de manera más precisa en la revisión de los obstáculos que impiden la conclusión de la transacción.

Historial de compras por Estado



Herramientas Utilizadas

MySQL Workbench

Microsoft Excel

PowerBI

Word

<https://www.dpriver.com/pp/sqlformat.htm>

<https://goonlinetools.com/word-to-markdown/>