UNIVERSIDAD AUTÓNOMA METROPOLITANA
Casa abierta al tiempo
Unidad Cuajimalpa

DCCD

PROYECTO DE BASES DE DATOS:

Sistema de Gestión de Tienda Online

MANUEL ALEJANDRO COLINDRES BARRIOS
PROFESOR: GUILLERMO MONROY RORIGUEZ
TRIMESTRE: 25 P
MATRICULA: 2233030327

INTRODUCCIÓN

En la época actual, han surgido nuevas tecnologías que facilitan tanto a personas como a empresas, sus trabajos. Al ir adaptando estas tecnologías en nuestra vida diaria, también van surgiendo nuevas necesidades.

En este proyecto nos vamos a enfocar en una necesidad que tienen los comercios, que son las tiendas digitales, para ello vamos a crear una base de datos eficiente y bien estructura que ayude a garantizar un rendimiento optimo en las necesidades que enfrenta nuestra tienda, como lo son la gestión de productos, reseñas y clientes.

OBJETIVOS

1. Estructura una base de datos relacional que nos ayude a gestionar productos, pedidos, clientes, reseñas de manera eficiente
2. Optimizar consultas mediante la MySQL para obtener información importante sobre los detalles de los productos, pedidos, clientes, etc.
3. Implementar procedimientos almacenados que ayuden a automatizar procesos importantes como lo es, el eliminar reseñas, agregar clientes, registrar pedidos, etc.
4. Ayuda a garantizar la escalabilidad del proyecto y mejorar su rendimiento mediante el uso de índices, normalización y buenas prácticas en la creación de la base de datos.

Diagrama ER y justificación de normalización.

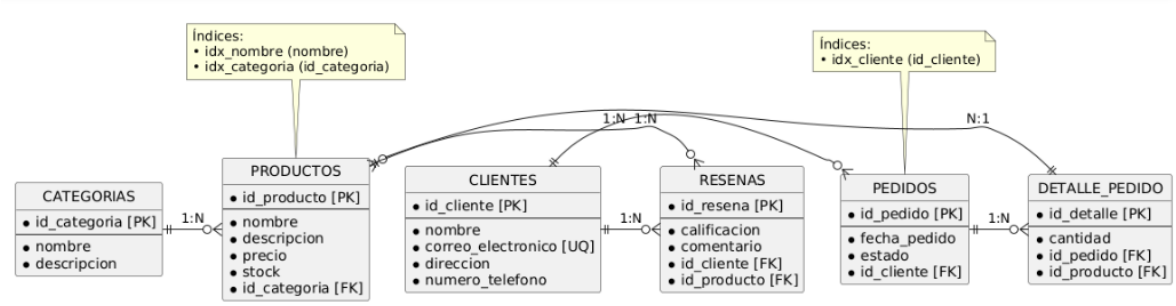

Ilustración 1: DIAGRAME ER HECHO MEIDANTE PLANT UML

Justificación de la Normalización

1. Primera Forma Normal (1NF):
   - Todas las tablas tienen atributos atómicos, sin listas ni datos repetidos. Por ejemplo, en Detalles_Pedido.
   - Se eliminan dependencias parciales y transitivas en los siguientes pasos.
2. Segunda Forma Normal (2NF):
   - Todas las tablas están en 1NF.
   - No hay dependencias parciales de claves primarias compuestas porque las claves primarias son simples, por ejemplo: id_detalle, id_resena.
   - Atributos como el precio en la tabla de Productos dependen por completo de la clave primaria.
3. Tercera Forma Normal (3NF):
   - No existen dependencias transitivas. Por ejemplo, en Productos, id_categoria no se determina otros atributos no clave como precio, solo lo relaciona con Categorías.
   - Se asegura que cada no clave dependa solo de la clave primaria.

Scripts SQL completos

-TABLAS E INDICES

```
CREATE DATABASE tienda_digital;
USE tienda_digital;

CREATE TABLE Clientes (
  id_cliente INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100),
  correo_electronico VARCHAR(100) UNIQUE,
  □eléfono□ VARCHAR(200),
  numero_telefono VARCHAR(30)
);

CREATE TABLE Categorias (
  id_categoria INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(200),
  □eléfono□□n VARCHAR(500)
);

CREATE TABLE Productos (
  id_producto INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100),
  □eléfono□□n VARCHAR(500),
  precio DECIMAL(10, 2),
  stock INT,
  id_categoria INT,
  FOREIGN KEY (id_categoria) REFERENCES Categorias(id_categoria)
);

CREATE TABLE Pedidos (
```

```
  id_pedido INT AUTO_INCREMENT PRIMARY KEY,
  id_cliente INT,
  estado VARCHAR(50),
  fecha_pedido DATE,
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)
);

CREATE TABLE Detalles_Pedido (
  id_detalle INT AUTO_INCREMENT PRIMARY KEY,
  id_pedido INT,
  id_producto INT,
  cantidad INT,
  FOREIGN KEY (id_pedido) REFERENCES Pedidos(id_pedido),
  FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)
);

CREATE TABLE Resenas (
  id_resena INT AUTO_INCREMENT PRIMARY KEY,
  id_cliente INT,
  id_producto INT,
  □eléfono□□n□ INT,
  comentario VARCHAR(500),
  FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente),
  FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)
);

CREATE INDEX idx_nombre ON Productos(nombre);
CREATE INDEX idx_categoria ON Productos(id_categoria);
CREATE INDEX idx_cliente ON Pedidos(id_cliente);
```

- **DATOS**

-- Insertar categorías

INSERT INTO Categorias (nombre, □eléfono□□n) VALUES

(‘Teléfonos’, ‘Smartphones y equipos móviles’),

(‘Laptops’, ‘Computadoras portátiles’),

(‘Accesorios’, ‘Periféricos y complementos’);

-- Insertar clientes

INSERT INTO Clientes (nombre, correo_electronico, □eléfono□, numero_telefono) VALUES

(‘Juan Pérez’, ‘juan.perez@gmail.com’, ‘Av. Insurgentes Sur 123’, ‘5512340101’),

(‘María López’, ‘maria.lopez@gmail.com’, ‘Paseo de la Reforma 456’, ‘5512340102’),

(‘Carlos Gómez’, ‘carlos.gomez@gmail.com’, ‘Plaza de la Constitución 789’, ‘5512340103’),

(‘Ana Martínez’, ‘ana.martinez@gmail.com’, ‘Eje Central Lázaro Cárdenas 321’, ‘5512340104’),

(‘Pedro Sánchez’, ‘pedro.sanchez@gmail.com’, ‘Calzada de Tlalpan 654’, ‘5512340105’),

(‘Laura Díaz’, ‘laura.diaz@gmail.com’, ‘Plaza Garibaldi 987’, ‘5512340106’),

(‘José Ramírez’, ‘jose.ramirez@gmail.com’, ‘Av. Río Churubusco 147’, ‘5512340107’),

(‘Sofía Hernández’, ‘sofia.hernandez@gmail.com’, ‘Av. Universidad 258’, ‘5512340108’),

(‘Miguel Torres’, ‘miguel.torres@gmail.com’, ‘Plaza Coyoacán 369’, ‘5512340109’),

(‘Lucía Castro’, ‘lucia.castro@gmail.com’, ‘Calle Génova 741, Zona Rosa’, ‘5512340110’),

(‘Fernando Ruiz’, ‘fernando.ruiz@gmail.com’, ‘Av. Patriotismo 852’, ‘5512340111’),

(‘Elena Vargas’, ‘elena.vargas@gmail.com’, ‘Plaza San Jacinto 963, San Ángel’, ‘5512340112’),

(‘Diego Morales’, ‘diego.morales@gmail.com’, ‘Av. División del Norte 159’, ‘5512340113’),

(‘Patricia Ortiz’, ‘patricia.ortiz@gmail.com’, ‘Av. Revolución 357’, ‘5512340114’),

(‘Javier Soto’, ‘javier.soto@gmail.com’, ‘Plaza Loreto 753, Altavista’, ‘5512340115’);

-- Insertar productos

INSERT INTO Productos (nombre, □eléfono□□n, precio, stock, id_categoria) VALUES

(‘iPhone 14’, ‘Smartphone de alta gama con pantalla Super Retina XDR, chip A15 Bionic, cámara dual de 12 MP, Face ID y resistencia al agua.’, 16999.83, 50, 1),

(‘Samsung Galaxy S23’, ‘Teléfono Android de última generación con pantalla AMOLED, procesador Snapdragon 8 Gen 2 y cámara de 50 MP.’, 15299.83, 45, 1),

(‘MacBook Air’, ‘Laptop ligera con chip Apple M2, pantalla Retina de 13.6”, batería de larga duración y diseño ultra delgado.’, 22099.83, 30, 2),

(‘Dell XPS 13’, ‘Laptop potente con procesador Intel Core i7, pantalla InfinityEdge, SSD de 512 GB y cuerpo de aluminio.’, 20399.83, 25, 2),

(‘Auriculares Bluetooth’, ‘Accesorio inalámbrico con micrófono integrado, cancelación de ruido pasiva y batería de hasta 10 horas.’, 849.83, 100, 3),

(‘Teclado Mecánico’, ‘Teclado para gaming con retroiluminación RGB, switches mecánicos y diseño ergonómico resistente al desgaste.’, 1359.83, 80, 3),

(‘iPhone 13’, ‘Smartphone anterior de Apple con excelente rendimiento, cámara dual y pantalla OLED de 6.1”.’, 13599.83, 40, 1),

(‘Lenovo ThinkPad’, ‘Laptop empresarial confiable con procesador Intel, teclado resistente a derrames y características de seguridad avanzadas.’, 18699.83, 20, 2),

(‘Mouse inalámbrico’, ‘Mouse ergonómico con conexión inalámbrica 2.4GHz, diseño cómodo y sensor óptico preciso.’, 509.83, 120, 3),

(‘Huawei P50’, ‘Teléfono avanzado con cámara Leica, pantalla OLED de 6.5”, y excelente rendimiento en fotografía y video.’, 11899.83, 35, 1),

(‘Asus ROG’, ‘Laptop gaming con tarjeta gráfica RTX, pantalla de alta tasa de refresco y sistema de enfriamiento avanzado.’, 25499.83, 15, 2),

(‘Cargador portátil’, ‘Cargador de 20W compatible con carga rápida, diseño compacto y seguro para dispositivos móviles.’, 679.83, 90, 3),

(‘Google Pixel 7’, ‘Teléfono con IA de Google, cámara avanzada, sistema Android puro y excelente rendimiento en fotografía nocturna.’, 15299.83, 30, 1),

(‘HP Spectre’, ‘Laptop ultradelgada con diseño premium, pantalla táctil, lector de huella y batería de larga duración.’, 22099.83, 20, 2),

(‘Funda protectora’, ‘Funda para teléfonos con material de silicona resistente, diseño antideslizante y bordes reforzados.’, 339.83, 150, 3),

(‘OnePlus 11’, ‘Teléfono rápido con carga ultra rápida, procesador Snapdragon 8 Gen 2 y pantalla AMOLED de 120Hz.’, 13599.83, 25, 1),

(‘Acer Nitro’, ‘Laptop gaming con tarjeta NVIDIA, procesador Ryzen y teclado retroiluminado para juegos intensos.’, 16999.83, 18, 2),

(‘Cable USB-C’, ‘Cable de carga trenzado, compatible con carga rápida y sincronización de datos.’, 254.83, 200, 3),

(‘Sony Xperia’, ‘Teléfono premium con pantalla 4K, sonido Hi-Res, cámara con óptica Zeiss y diseño elegante.’, 16149.83, 22, 1),

(‘Microsoft Surface’, ‘Laptop 2 en 1 con pantalla táctil PixelSense, lápiz óptico y teclado desmontable.’, 23799.83, 15, 2),

(‘Altavoz Bluetooth’, ‘Altavoz portátil con sonido estéreo, resistencia al agua IPX5 y batería recargable de larga duración.’, 1019.83, 70, 3),

(‘Xiaomi 13’, ‘Teléfono económico con gran rendimiento, pantalla AMOLED y sistema de cámaras triple.’, 10199.83, 40, 1),

(‘Razer Blade’, ‘Laptop de alto rendimiento para gaming, chasis de aluminio, pantalla QHD y GPU dedicada.’, 30599.83, 10, 2),

(‘Soporte para teléfono’, ‘Soporte ajustable con base antideslizante, compatible con la mayoría de smartphones.’, 424.83, 110, 3),

(‘Oppo Find X5’, ‘Teléfono con cámara de alto nivel, diseño elegante y pantalla AMOLED con alta tasa de refresco.’, 12749.83, 28, 1),

(‘Toshiba Satellite’, ‘Laptop básica ideal para tareas escolares y de oficina, con almacenamiento HDD y pantalla HD.’, 11899.83, 25, 2),

(‘Batería externa’, ‘Batería de 10000mAh con doble salida USB y linterna LED, ideal para viajes.’, 594.83, 85, 3),

(‘Realme GT’, ‘Teléfono de gama media con gran potencia, buena autonomía y pantalla fluida de 120Hz.’, 8499.83, 35, 1),

(‘LG Gram’, ‘Laptop ultraligera con diseño minimalista, batería de larga duración y pantalla de 14”.’, 20399.83, 20, 2),

(‘Auriculares con micrófono’, ‘Auriculares económicos con cable, micrófono integrado y diseño cómodo para uso prolongado.’, 509.83, 95, 3);

-- Insertar pedidos

INSERT INTO Pedidos (id_cliente, estado, fecha_pedido) VALUES

(1, ‘pendiente’, ‘2025-07-30’),

(1, ‘enviado’, ‘2025-07-25’),

(2, ‘entregado’, ‘2025-07-20’),

(3, ‘pendiente’, ‘2025-07-15’),

(4, ‘enviado’, ‘2025-07-10’),

(5, ‘entregado’, ‘2025-07-05’),

(6, ‘pendiente’, ‘2025-06-30’),

(7, ‘enviado’, ‘2025-06-27’),

(8, ‘entregado’, ‘2025-06-21’),

(9, ‘pendiente’, ‘2025-06-18’),

(10, ‘enviado’, ‘2025-06-13’),

(11, ‘entregado’, ‘2025-06-10’),

(12, ‘pendiente’, ‘2025-06-05’),

(13, ‘enviado’, ‘2025-06-01’),

(14, ‘entregado’, ‘2025-05-28’),

(15, ‘pendiente’, ‘2025-05-24’),

(1, ‘enviado’, ‘2025-05-20’),

(2, ‘entregado’, ‘2025-05-15’),

(3, ‘pendiente’, ‘2025-05-10’),

(4, ‘enviado’, ‘2025-05-05’);

-- Insertar detalles de pedido

INSERT INTO Detalles_Pedido (id_pedido, id_producto, cantidad) VALUES

(1, 1, 1),

(2, 2, 1),

(3, 3, 1),

(4, 4, 1),

(5, 5, 2),

(6, 6, 1),

(7, 7, 1),

(8, 8, 1),

(9, 9, 2),

(10, 10, 1),

(11, 11, 1),

(12, 12, 2),

(13, 13, 1),

(14, 14, 1),

(15, 15, 1),

(16, 16, 1),

(17, 17, 1),

(18, 18, 2),

(19, 19, 1),

(20, 20, 1),

(1, 21, 1),

(2, 22, 1),

(3, 23, 1),

(4, 24, 2),

(5, 25, 1);

-- Insertar reseñas

INSERT INTO Resenas (id_cliente, id_producto, □eléfono□□n□, comentario) VALUES

(1, 1, 5, ‘Excelente producto’),

(2, 2, 4, ‘Muy bueno’),

(3, 3, 5, ‘Increíble calidad’),

(4, 4, 3, ‘Regular’),

(5, 5, 4, ‘Buen accesorio’),

(6, 6, 5, ‘Perfecto’),

(7, 7, 4, ‘Satisfecho’),

(8, 8, 3, ‘Aceptable’),

(9, 9, 5, ‘Gran compra’),

(10, 10, 4, ‘Bueno’);

- CONSULTAS Y PROCEDIMIENTOS

-- 1. Listar productos disponibles por categoría, ordenados por precio.

-- Se filtran productos con stock > 0, se unen con su categoría y se ordenan por categoría y precio.

SELECT c.nombre AS □eléfono□, p.nombre, p.precio

FROM Productos p

INNER JOIN Categorias c ON p.id_categoria = c.id_categoria

WHERE p.stock > 0

ORDER BY c.nombre, p.precio ASC;

-- 2. Mostrar clientes con pedidos pendientes y total de compras.

-- Se consideran solo pedidos con estado ‘pendiente’, se agrupan por cliente y se calcula el total gastado.

SELECT cl.id_cliente, cl.nombre, COUNT(p.id_pedido) AS pedidos_pendientes,

SUM(pd.precio * dp.cantidad) AS total_compras

FROM Clientes cl

INNER JOIN Pedidos p ON cl.id_cliente = p.id_cliente

INNER JOIN Detalles_Pedido dp ON p.id_pedido = dp.id_pedido

INNER JOIN Productos pd ON dp.id_producto = pd.id_producto

WHERE p.estado = ‘pendiente’

GROUP BY cl.id_cliente, cl.nombre;

-- 3. Reporte de los 5 productos con mejor calificación promedio en reseñas.

-- Se calculan los promedios de calificaciones por producto(si tienen reseñas) y se seleccionan los 5 más altos.

SELECT p.id_producto, p.nombre, AVG(r.calificacion) AS promedio

FROM Productos p

INNER JOIN Resenas r ON p.id_producto = r.id_producto

GROUP BY p.id_producto, p.nombre

ORDER BY promedio DESC

LIMIT 5;

--PROCEDIMIENTOS ALMACENADOS

-- 1. Registrar un nuevo pedido verificando el límite de 5 pedidos pendientes y stock suficiente.

```
DELIMITER $$

CREATE PROCEDURE RegistrarPedido (

    IN p_id_cli INT,       -- ID del cliente que realiza el pedido

    IN p_id_prod INT,      -- ID del producto solicitado

    IN p_cantidad INT      -- Cantidad deseada del producto

)

BEGIN

    -- Se hace un □eléfono de pedidos pendientes, verifica que el cliente no tenga más de 5 pedidos pendientes asi se evita la acumulación excesiva de pedidos sin procesar

    IF (SELECT COUNT(*) FROM Pedidos WHERE id_cliente = p_id_cli AND estado = 'pendiente') >= 5 THEN

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Máximo 5 pedidos pendientes permitidos';

    END IF;


    -- Disponibilidad de stock: Comprueba que haya suficiente inventario para satisfacer el pedido

    IF (SELECT stock FROM Productos WHERE id_producto = p_id_prod) < p_cantidad THEN

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Stock insuficiente';

    END IF;


    /*

    Creación del pedido

    Inserta un nuevo registro en la tabla Pedidos con:

    - ID del cliente

    - Estado inicial 'pendiente'

    - Fecha actual automática

    */

    INSERT INTO Pedidos(id_cliente, estado, fecha_pedido)

    VALUES (p_id_cli, 'pendiente', NOW());


    /*

    Detalle del pedido
```

```
    Inserta en Detalles_Pedido la información específica:
    - ID del pedido recién creado (LAST_INSERT_ID())
    - ID del producto solicitado
    - Cantidad requerida
    */
    INSERT INTO Detalles_Pedido(id_pedido, id_producto, cantidad)
    VALUES (LAST_INSERT_ID(), p_id_prod, p_cantidad);

    /*
    Reduce el stock disponible del producto según la cantidad pedida
    */
    UPDATE Productos SET stock = stock – p_cantidad WHERE id_producto = p_id_prod;

    -- Mensaje de confirmación de éxito
    SELECT 'Pedido registrado correctamente' AS Mensaje;
END $$
DELIMITER ;
```

-- 2. Registrar una reseña, verificando que el cliente haya comprado el producto.

```
DELIMITER $$
CREATE PROCEDURE RegistrarResena (
    IN p_id_cli INT,          -- ID del cliente que escribe la reseña
    IN p_id_prod INT,         -- ID del producto evaluado
    IN p_calificacion INT,    -- Puntuación dada (1-5)
    IN p_comentario VARCHAR(500) – Texto de la reseña
)
BEGIN
    -- Asegura que la puntuación esté entre 1 y 5 estrellas
    IF p_calificacion < 1 OR p_calificacion > 5 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La calificación debe ser entre 1 y 5';
    END IF;

    /*
    Verifica que el cliente haya comprado efectivamente el producto
    Previene reseñas falsas o no verificadas
```

```
   */
   IF NOT EXISTS (
      SELECT 1
      FROM Pedidos p
      INNER JOIN Detalles_Pedido dp ON p.id_pedido = dp.id_pedido
      WHERE p.id_cliente = p_id_cli AND dp.id_producto = p_id_prod
   ) THEN
      SIGNAL SQLSTATE '45000'
      SET MESSAGE_TEXT = 'El cliente no ha comprado el producto';
   END IF;
   -- Registra la evaluación con todos los datos proporcionados:
   INSERT INTO Resenas(id_cliente, id_producto, □eléfono□□n□, comentario)
   VALUES (p_id_cli, p_id_prod, p_calificacion, p_comentario);

   -- Mensaje de confirmación
   SELECT 'Reseña registrada correctamente' AS Mensaje;
END $$
DELIMITER ;
```

-- 3. Actualizar el stock de un producto después de un pedido.

```
DELIMITER $$
CREATE PROCEDURE ActualizarStock (
   IN p_id_prod INT,      -- ID del producto a actualizar
   IN p_cantidad INT      -- Cantidad a reducir del stock
)
BEGIN
   -- Se compara la cantidad solicitada con el inventario actual
   IF (SELECT stock FROM Productos WHERE id_producto = p_id_prod) < p_cantidad THEN
      SELECT 'Error: No hay suficiente stock' AS Mensaje;
   ELSE
   -- Se hace la resta al stock solo si hay disponibilidad
      UPDATE Productos SET stock = stock – p_cantidad WHERE id_producto = p_id_prod;
      SELECT 'Stock actualizado correctamente' AS Mensaje;
   END IF;
END $$
```

```
DELIMITER ;
```

-- 4. Cambiar el estado de un pedido (por ejemplo, de pendiente a enviado).

```
DELIMITER $$
CREATE PROCEDURE CambiarEstadoPedido (
    IN p_id_ped INT,              -- ID del pedido a modificar
    IN p_nuevo_estado VARCHAR(50)   -- Nuevo estado a asignar
)
BEGIN

    -- Se verifica que el pedido no tenga el estado que se quiere asignar asi se evitan actualizaciones innecesarias
    IF (SELECT estado FROM Pedidos WHERE id_pedido = p_id_ped) = p_nuevo_estado THEN
        SELECT CONCAT('Error: El pedido ya está en estado ', p_nuevo_estado) AS Mensaje;
    ELSE
        -- Modifica el estado del pedido al nuevo valor proporcionado
        UPDATE Pedidos SET estado = p_nuevo_estado WHERE id_pedido = p_id_ped;
        -- ROW_COUNT() retorna el número de filas afectadas, si es 0 significa que no se encontró el pedido.
        IF ROW_COUNT() = 0 THEN
            SELECT 'Error: No se encontró el pedido' AS Mensaje;
        ELSE
            SELECT CONCAT('Estado actualizado a: ', p_nuevo_estado) AS Mensaje;
        END IF;
    END IF;
END $$
DELIMITER ;
```

-- 5. Eliminar reseñas de un producto específico.

```
DELIMITER $$
CREATE PROCEDURE EliminarResenasProducto (
    IN p_id_prod INT    -- ID del producto cuyas reseñas se eliminarán
)
BEGIN
    -- Verifica que el producto exista en la base de datos
    IF NOT EXISTS (SELECT 1 FROM Productos WHERE id_producto = p_id_prod) THEN
        SELECT 'Error: El producto no existe' AS Mensaje;
    -- Comprueba que el producto tenga reseñas antes de intentar borrar
```

```
  ELSEIF NOT EXISTS (SELECT 1 FROM Resenas WHERE id_producto = p_id_prod) THEN
    SELECT 'Advertencia: El producto no tiene reseñas' AS Mensaje;
  ELSE
    -- Borra todas las evaluaciones asociadas al producto
    DELETE FROM Resenas WHERE id_producto = p_id_prod;
    /*
    RESULTADO Y CALIFICACIÓN PROMEDIO
    Muestra:
    1. Cantidad de reseñas eliminadas
    2. Nuevo promedio (5 por defecto ya que no hay reseñas)
    */
    SELECT
      CONCAT(ROW_COUNT(), ' reseña(s) eliminada(s)') AS Mensaje,
      IFNULL(
        (SELECT AVG(□eléfono□□n□)
         FROM Resenas
         WHERE id_producto = p_id_prod),
        5
      ) AS 'Nueva calificación';
  END IF;
END $$
DELIMITER ;
```

-- 6. Agregar un nuevo producto, verificando que no exista un duplicado (mismo nombre y categoría).

```
DELIMITER $$
CREATE PROCEDURE AgregarProducto (
  IN p_nombre VARCHAR(100),   -- Nombre del nuevo producto
  IN p_id_cat INT,            -- ID de la categoría del producto
  IN p_desc VARCHAR(500),     -- Descripción detallada
  IN p_precio DECIMAL(10,2),  -- Precio unitario
  IN p_stock INT              -- Cantidad inicial en inventario
)
BEGIN
  -- Verifica que no exista otro producto con el mismo nombre en la misma categoría
  IF EXISTS (SELECT 1 FROM Productos WHERE nombre = p_nombre AND id_categoria = p_id_cat) THEN
    SELECT 'Error: Producto duplicado (mismo nombre y categoría)' AS Mensaje;
```

```
    -- Confirma que la categoría especificada exista en la base de datos
    ELSEIF NOT EXISTS (SELECT 1 FROM Categorias WHERE id_categoria = p_id_cat) THEN
        SELECT 'Error: La categoría no existe' AS Mensaje;
    -- Asegura que el precio sea un valor positivo
    ELSEIF p_precio <= 0 THEN
        SELECT 'Error: El precio debe ser positivo' AS Mensaje;
    -- Verifica que el stock inicial no sea negativo
    ELSEIF p_stock < 0 THEN
        SELECT 'Error: El stock no puede ser negativo' AS Mensaje;

    ELSE
        /*
        SE INSERTA EL PRODUCTO
        Si pasó todas las validaciones, crea el nuevo registro con:
        - Nombre
                                                    - Categoría
        - Descripción
        - Precio
        - Stock inicial
        */
        INSERT INTO Productos(nombre, id_categoria, □eléfono□□n, precio, stock)
        VALUES (p_nombre, p_id_cat, p_desc, p_precio, p_stock);

        -- Mensaje de confirmación
        SELECT 'Producto agregado correctamente' AS Mensaje;
    END IF;
END $$
DELIMITER ;
```

-- 7. Actualizar la información de un cliente (por ejemplo, dirección o teléfono).

```
DELIMITER $$
CREATE PROCEDURE ActualizarCliente (
    IN p_id_cli INT,            -- ID del cliente a actualizar
    IN p_direccion VARCHAR(200), -- Nueva dirección
    IN p_telefono VARCHAR(30)    -- Nuevo teléfono
)
```

```
BEGIN
   -- Verifica que el cliente exista antes de actualizar
   IF NOT EXISTS (SELECT 1 FROM Clientes WHERE id_cliente = p_id_cli) THEN
      SELECT ‘Error: El cliente no existe’ AS Mensaje;
   ELSE
      /*
      Modifica solo los campos proporcionados:
      - Dirección
      - Teléfono
      */
      UPDATE Clientes
      SET □eléfono□ = p_direccion,
         numero_telefono = p_telefono
      WHERE id_cliente = p_id_cli;
      -- ROW_COUNT() indica si hubo modificaciones
      IF ROW_COUNT() > 0 THEN
         SELECT ‘Datos del cliente actualizados correctamente’ AS Mensaje;
      ELSE
         -- Si no hubo cambios, informa que los datos eran iguales
         SELECT ‘Advertencia: No se realizaron cambios (los datos son iguales)’ AS Mensaje;
      END IF;
   END IF;
END $$
DELIMITER ;
```

-- 8. Generar un reporte de productos con stock bajo (menos de 5 unidades).

```
DELIMITER $$
CREATE PROCEDURE ReporteStockBajo ()
BEGIN
   -- Comprueba si existen productos con menos de 5 unidades
   IF EXISTS (SELECT 1 FROM Productos WHERE stock < 5) THEN
      /*
      Lista todos los productos con stock bajo
      - ID del producto
      - Nombre
      - Cantidad actual
```

```
    Ordenados de menor a mayor stock
    */
    SELECT id_producto, nombre, stock
    FROM Productos
    WHERE stock < 5
    ORDER BY stock ASC;
    -- Muestra el conteo total de productos con stock bajo
    SELECT CONCAT('Se encontraron ', COUNT(*), ' productos con stock bajo') AS Resumen
    FROM Productos WHERE stock < 5;
  ELSE
    -- Mensaje informativo cuando no hay stock bajo
    SELECT 'No hay productos con stock bajo (menos de 5 unidades)' AS Mensaje;
  END IF;
END $$
DELIMITER ;
```

Resultados de pruebas y análisis de rendimiento.

```
mysql> SELECT c.nombre AS categoria, p.nombre, p.precio
    -> FROM Productos p
    -> INNER JOIN Categorias c ON p.id_categoria = c.id_categoria
    -> WHERE p.stock > 0
    -> ORDER BY c.nombre, p.precio ASC;
+------------+----------------------------+----------+
| categoria  | nombre                     | precio   |
+------------+----------------------------+----------+
| Accesorios | Cable USB-C                |   254.83 |
| Accesorios | Funda protectora           |   339.83 |
| Accesorios | Soporte para tel├@fono     |   424.83 |
| Accesorios | Mouse inal├ímbrico         |   509.83 |
| Accesorios | Auriculares con micr├│fono |   509.83 |
| Accesorios | Bater├¡a externa           |   594.83 |
| Accesorios | Cargador port├átil         |   679.83 |
| Accesorios | Auriculares Bluetooth      |   849.83 |
| Accesorios | Altavoz Bluetooth          |  1019.83 |
| Accesorios | Teclado Mec├ínico          |  1359.83 |
| Laptops    | Toshiba Satellite          | 11899.83 |
| Laptops    | Acer Nitro                 | 16999.83 |
| Laptops    | Lenovo ThinkPad            | 18699.83 |
| Laptops    | Dell XPS 13                | 20399.83 |
| Laptops    | LG Gram                    | 20399.83 |
| Laptops    | MacBook Air                | 22099.83 |
| Laptops    | HP Spectre                 | 22099.83 |
| Laptops    | Microsoft Surface          | 23799.83 |
| Laptops    | Asus ROG                   | 25499.83 |
| Laptops    | Razer Blade                | 30599.83 |
| Tel├@fonos | Realme GT                  |  8499.83 |
| Tel├@fonos | Xiaomi 13                  | 10199.83 |
| Tel├@fonos | Huawei P50                 | 11899.83 |
| Tel├@fonos | Oppo Find X5               | 12749.83 |
| Tel├@fonos | iPhone 13                  | 13599.83 |
| Tel├@fonos | OnePlus 11                 | 13599.83 |
| Tel├@fonos | Samsung Galaxy S23         | 15299.83 |
| Tel├@fonos | Google Pixel 7             | 15299.83 |
| Tel├@fonos | Sony Xperia                | 16149.83 |
| Tel├@fonos | iPhone 14                  | 16999.83 |
+------------+----------------------------+----------+
30 rows in set (0.01 sec)
```

```
mysql> SELECT cl.id_cliente, cl.nombre, COUNT(p.id_pedido) AS pedidos_pendientes,
    ->        SUM(pd.precio * dp.cantidad) AS total_compras
    -> FROM Clientes cl
    -> INNER JOIN Pedidos p ON cl.id_cliente = p.id_cliente
    -> INNER JOIN Detalles_Pedido dp ON p.id_pedido = dp.id_pedido
    -> INNER JOIN Productos pd ON dp.id_producto = pd.id_producto
    -> WHERE p.estado = 'pendiente'
    -> GROUP BY cl.id_cliente, cl.nombre;
+------------+--------------+--------------------+----------------+
| id_cliente | nombre       | pedidos_pendientes | total_compras  |
+------------+--------------+--------------------+----------------+
|          1 | Juan P├@rez  |                  2 |       18019.66 |
|          3 | Carlos G├│mez |                  3 |       37399.32 |
|          6 | Laura D├¡az  |                  1 |       13599.83 |
|          9 | Miguel Torres |                 1 |       11899.83 |
|         12 | Elena Vargas |                  1 |       15299.83 |
|         15 | Javier Soto  |                  1 |       13599.83 |
+------------+--------------+--------------------+----------------+
6 rows in set (0.01 sec)
```

```
mysql> SELECT p.id_producto, p.nombre, AVG(r.calificacion) AS promedio
    -> FROM Productos p
    -> INNER JOIN Resenas r ON p.id_producto = r.id_producto
    -> GROUP BY p.id_producto, p.nombre
    -> ORDER BY promedio DESC
    -> LIMIT 5;
+-------------+--------------------+----------+
| id_producto | nombre             | promedio |
+-------------+--------------------+----------+
|           1 | iPhone 14          |   5.0000 |
|           3 | MacBook Air        |   5.0000 |
|           9 | Mouse inal├ímbrico |   5.0000 |
|           6 | Teclado Mec├ínico  |   5.0000 |
|           2 | Samsung Galaxy S23 |   4.0000 |
+-------------+--------------------+----------+
5 rows in set (0.01 sec)
```

```
mysql> EXPLAIN SELECT * FROM Productos WHERE nombre LIKE 'iPhone%';
+----+-------------+-----------+------------+-------+---------------+------------+---------+------+------+----------+-----------------------+
| id | select_type | table     | partitions | type  | possible_keys | key        | key_len | ref  | rows | filtered | Extra                 |
+----+-------------+-----------+------------+-------+---------------+------------+---------+------+------+----------+-----------------------+
|  1 | SIMPLE      | Productos | NULL       | range | idx_nombre    | idx_nombre | 403     | NULL |    2 |   100.00 | Using index condition |
+----+-------------+-----------+------------+-------+---------------+------------+---------+------+------+----------+-----------------------+
1 row in set, 1 warning (0.01 sec)
```

MANUEL ALEJANDRO COLINDRES BARRIOS 25 P
2233030327

```
mysql> CALL RegistrarPedido(1, 1, 1);
+----------------------------------+
| Mensaje                          |
+----------------------------------+
| Pedido registrado correctamente  |
+----------------------------------+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CALL RegistrarResena(1, 1, 5, 'Excelente producto');
+----------------------------------+
| Mensaje                          |
+----------------------------------+
| Reseña registrada correctamente  |
+----------------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)
```

```
mysql> CALL RegistrarPedido(1, 1, 100);
ERROR 1644 (45000): Stock insuficiente
```

```
mysql> CALL RegistrarResena(1, 1, 6, 'Demasiado bueno');
ERROR 1644 (45000): La calificación debe ser entre 1 y 5
```

```
mysql> CALL ActualizarStock(5, 2);
+----------------------------------+
| Mensaje                          |
+----------------------------------+
| Stock actualizado correctamente  |
+----------------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> SELECT stock FROM Productos WHERE id_producto = 5;
+-------+
| stock |
+-------+
|    96 |
+-------+
1 row in set (0.00 sec)
```

```
mysql> CALL CambiarEstadoPedido(2, 'enviado');
+-----------------------------------------+
| Mensaje                                 |
+-----------------------------------------+
| Error: El pedido ya está en estado enviado |
+-----------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CALL EliminarResenasProducto(5);
+-------------------------+--------------------+
| Mensaje                 | Nueva calificación |
+-------------------------+--------------------+
| 1 reseña(s) eliminada(s) |             5.0000 |
+-------------------------+--------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CALL ActualizarStock(5, 150);
+----------------------------------+
| Mensaje                          |
+----------------------------------+
| Error: No hay suficiente stock   |
+----------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CALL EliminarResenasProducto(30);
+-----------------------------------------+
| Mensaje                                 |
+-----------------------------------------+
| Advertencia: El producto no tiene reseñas |
+-----------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CALL AgregarProducto('Monitor Gaming Curvo Odyssey Neo G9', 3, 'Monitor super ultra-wide de 49" con resolución Dual QHD (5120x1440), relación de aspecto 32:9, curvatura 1000R, panel
 VA Quantum Mini-LED, frecuencia de actualización de 240Hz, tiempo de respuesta de 1ms GtG, compatibilidad con NVIDIA G-SYNC y AMD FreeSync Premium Pro, brillo HDR2000.',45999.99, 8);
+----------------------------------+
| Mensaje                          |
+----------------------------------+
| Producto agregado correctamente  |
+----------------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CALL AgregarProducto('iPhone 14', 1, 'Otra .....', 15000, 5);
+-----------------------------------------------------+
| Mensaje                                             |
+-----------------------------------------------------+
| Error: Producto duplicado (mismo nombre y categoria) |
+-----------------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CALL ActualizarCliente(1, 'Av. Reforma 500', '5511223344');
+-----------------------------------------+
| Mensaje                                 |
+-----------------------------------------+
| Datos del cliente actualizados correctamente |
+-----------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

```
mysql> CALL ActualizarCliente(999, 'Dirección', '5555555');
+-----------------------------+
| Mensaje                     |
+-----------------------------+
| Error: El cliente no existe |
+-----------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
mysql> UPDATE Productos SET stock = 3 WHERE id_producto = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> CALL ReporteStockBajo();
+-------------+-----------------------+-------+
| id_producto | nombre                | stock |
+-------------+-----------------------+-------+
|           5 | Auriculares Bluetooth |     3 |
+-------------+-----------------------+-------+
1 row in set (0.00 sec)

+-----------------------------------------+
| Resumen                                 |
+-----------------------------------------+
| Se encontraron 1 productos con stock bajo |
+-----------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Propuestas de Mejoras

- Índices adicionales:
    - CREATE INDEX idx_estado ON Pedidos(estado); - Para consultas por estado
    - CREATE INDEX idx_precio ON Productos(precio); - Para consultas ordenadas por precio
- Se podría agregar el procedimiento para eliminar un producto de la tienda.

Conclusiones y lecciones aprendidas.

Durante este proyecto, aprendí muchas cosas importantes:

1. Crear una base de datos eficientes, mediante el uso de tablas me permitió tener un proyecto bien organizado, estructurado y eficiente, para que sea más fácil gestionarlo, además el normalizar, me ayuda a evitar redundancias y mejora el como se ven los datos.
2. Aprendí que los stored procedures me ayudan a automatizar procesos clave para mejorar el rendimiento de mi base de datos, al pasar mediante ciertas validaciones, me ayuda a evitar tener errores.
3. Al crear índices, puedo acelerar ciertas búsquedas, haciendo más fácil el trabajo de encontrar algún dato en específico, aunque también en el apartado de mejoras, redacte que serían necesarios crear más.
4. El documentar todo mi trabajo, también me ayudo a encontrar ciertos errores, el saber donde se encontraba cada proceso, el dar mejoras a los códigos para evitar fallas, y para un futuro, si quiero implementar más cosas, puedo saber desde donde partir.