

BASE DE DATOS

EMPRENDIMIENTO
DE VENTA Y COMERCIO DE ESPECIAS Y
CONDIMENTOS

Jeremias Israel Candia

Descripción del negocio

Esta base de datos es creada con el propósito de organizar y almacenar datos esenciales que abarcan desde la gestión de productos de condimentos y especias hasta la interacción con los clientes y proveedores.

Esta base de datos es una herramienta vital para garantizar el éxito y la eficiencia en el emprendimiento de ventas de condimentos y especias. Facilita la gestión de productos, clientes, proveedores y ventas, proporcionando una visión integral de las operaciones y permitiendo la toma de decisiones informadas para el crecimiento del negocio.

Funcionalidades Principales

- Gestión de inventario en tiempo real, permitiendo un seguimiento preciso de los niveles de stock.
- Registro y seguimiento de ventas, lo que facilita la identificación de productos más populares y rentables.
- Facilita la gestión de proveedores y la optimización de relaciones comerciales.
- Registra el historial de cambios en el inventario para un control detallado de las operaciones.
- Generación de informes y estadísticas para la toma de decisiones basada en datos.

EMPRENDIMIENTO

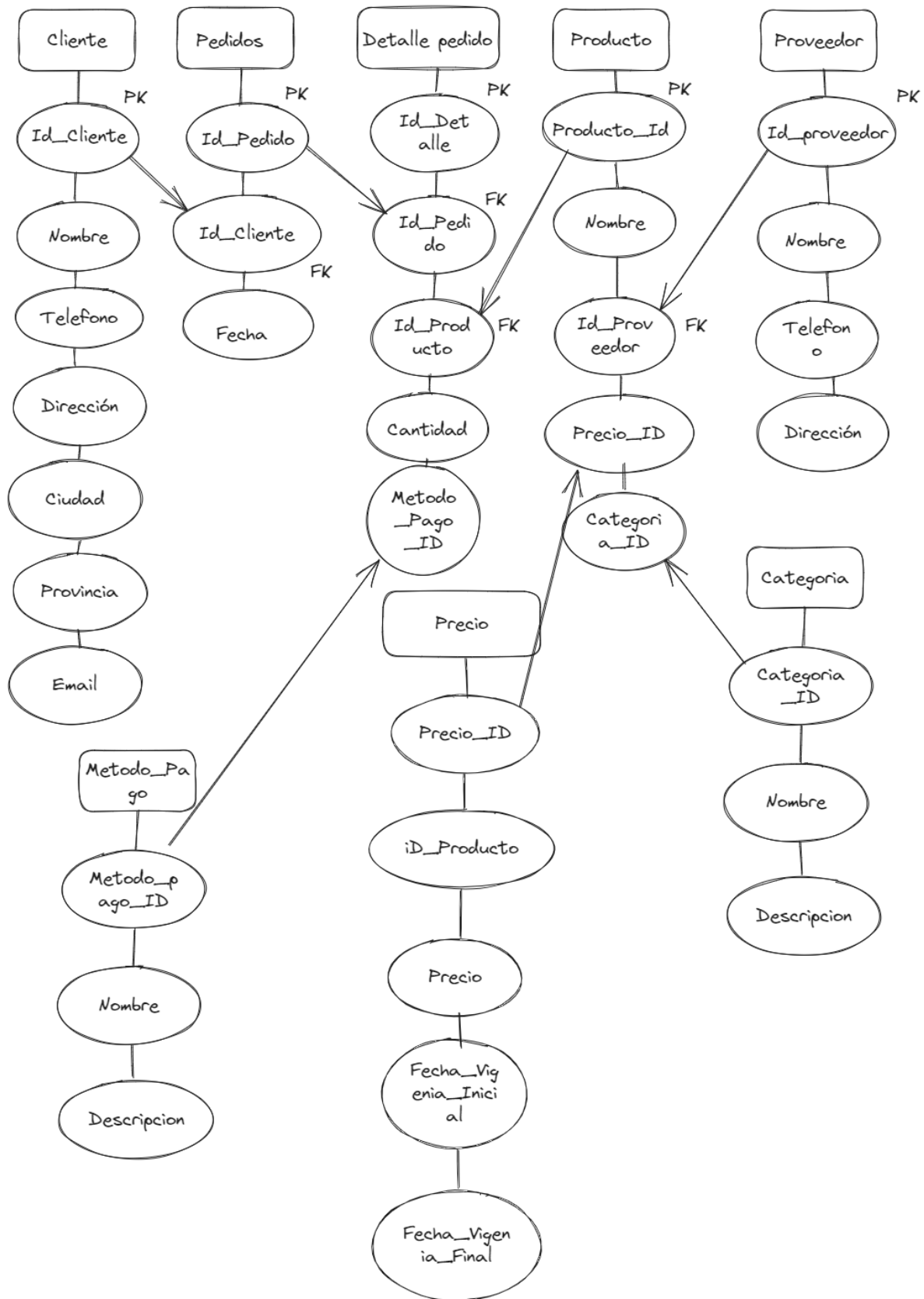


TABLA: Clientes		
NOMBRE	DESCRIPCION: Tabla referida con todos los datos del cliente	TIPO DE DATO
Cliente_ID	Primary key	INT
Nombre	Nombre del cliente	VARCHAR(50)
Telefono	Numero de telefono del cliente	VARCHAR(50)
Direccion	Domicilio del cliente	VARCHAR(50)
Ciudad	Localidad	VARCHAR(50)
Provincia	Provincia donde esta ubicado el cliente	VARCHAR(50)
Email	Email de contacto	VARCHAR(50)

TABLA: Empleados		
NOMBRE	DESCRIPCION: Tabla referida con los datos del empleado	TIPO DE DATO
Empleado_ID	Primary key	INT
Nombre_empleado	Nombre del empleado	VARCHAR(20)
Apellido	Apellido del empleado	VARCHAR(20)
Fecha_ingreso	Año, mes y día de ingreso del empleado al trabajo	DATE
Salario	Valor del salario del empleado en moneda nacional	DECIMAL(10,2)

TABLA: Proveedores		
NOMBRE	DESCRIPCION: Incluye todos los datos de los proveedores	TIPO DE DATO
Proveedor_ID	Primary key	INT
Nombre_Proveedor	Nombre del proveedor de condimentos y especias	VARCHAR(50)
Telefono	Numero de telefono de contacto	VARCHAR(50)
Direccion	Domicilio del proveedor	VARCHAR(50)
Ciudad	Localidad	VARCHAR(50)
Provincia	Provincia donde esta ubicado el cliente	VARCHAR(50)
Email	Email de contacto	VARCHAR(50)

TABLA: Categoria		
NOMBRE	DESCRIPCION: Detalles de que clases son los condimentos y especias	TIPO DE DATO
Categoria_ID	Primary key	INT
Nombre_categoria	Es el nombre del tipo de especia a la que pertenece el producto	VARCHAR(20)
Descripcion	Descripcion del condimento o especia	VARCHAR(255)

TABLA: Productos		
NOMBRE	DESCRIPCION: Detalle de los productos que se venden	TIPO DE DATO
Producto_ID	Primary key	INT
Nombre_Producto	Nombre del producto	VARCHAR(50)
Proveedor_ID	Foreign Key: Referencia a la tabla proveedores	INT
Categoria_ID	Foreign Key: Referencia a la tabla Categoria	INT

Precio_ID	Foreign key: Valor monetario del producto (El precio es por Kg)	FLOAT
-----------	---	-------

TABLA: Precio		
NOMBRE	DESCRIPCION: Valores monetarios de los productos	TIPO DE DATO
Precio_ID	Primary key	INT
Producto_ID	Foreign Key: Referencia a la tabla: Productos	INT
Precio_vigencia_inicial	Valor del producto inicial	FLOAT
Precio_vigencia_final	Valor del producto final	FLOAT

TABLA: Metodo_pago		
NOMBRE	DESCRIPCION: Especificación del metodo de pago de la compra	TIPO DE DATO
Metodo_pago_ID	Primary key	INT
Nombre_Pago	Que metodo de pago se realiza	VARCHAR(50)
Descripcion	Descripcion del forma de pago y de la compra	VARCHAR(100)

TABLA: Pedido		
NOMBRE	DESCRIPCION: Todos los pedidos registrados en la tabla	TIPO DE DATO
Pedido_ID	Primary key	INT
Cliente_ID	Foreign key: referencia al cliente que realizo la compra	INT
Empleado_ID	Foreign Key: Referencia al empleado que vendio el/los producto/s	INT
Fecha	Año, mes y dia que se realizó el pedido	DATE

TABLA: Detalle_pedido		
NOMBRE	DESCRIPCION: Tabla que detalla las ventas realizadas	TIPO DE DATO
Detalle_pedido_ID	Primary key	INT
Pedido_ID	Foreign key: referencia a la venta en la tabla Pedidos	INT
Producto_ID	Foreign Key: Referencia al producto vendido, encontrado en la tabla Productos	INT
Cantidad	Cantidad	INT
Metodo_Pago_ID	Foreign Key: Refiere a la tabla: metodo_pago	INT

SCRIPT .SQL DE LAS TABLAS

```
DROP DATABASE IF EXISTS Emprendimiento_Condimento;
CREATE DATABASE IF NOT EXISTS
Emprendimiento_Condimento;
USE Emprendimiento_Condimento;
```

```
-- creación de tabla categoria
```

```
CREATE TABLE Categoria
(
    Categoria_ID INT NOT NULL,
    Nombre_Categoria VARCHAR(50) NOT NULL,
    Descripcion VARCHAR(255) NOT NULL,
    PRIMARY KEY (Categoria_ID)
);
```

```
-- creación de tabla clientes
```

```
CREATE TABLE Clientes
(
    Cliente_ID INT NOT NULL,
    Nombre_cliente VARCHAR(50) NOT NULL,
    Telefono VARCHAR (50) NOT NULL,
    Direccion VARCHAR(50) NOT NULL,
    Ciudad VARCHAR(50) NOT NULL,
    Provincia VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
```

```

        PRIMARY KEY (Cliente_ID)
    );

-- creación de tabla empleados
CREATE TABLE Empleados
(
    Empleado_ID INT NOT NULL,
    Nombre_empleado VARCHAR(20) NOT NULL,
    Apellido VARCHAR(20) NOT NULL,
    Fecha_ingreso DATE NOT NULL,
    salario DECIMAL(10,2) NOT NULL,
    PRIMARY KEY(Empleado_ID)
);

-- tabla de los proveedores
CREATE TABLE Proveedores(
    Proveedor_ID INT NOT NULL,
    Nombre_proveedor VARCHAR(50) NOT NULL,
    Telefono VARCHAR(50) NOT NULL,
    Direccion VARCHAR(50) NOT NULL,
    Ciudad VARCHAR(25) NOT NULL,
    Provincia VARCHAR(25) NOT NULL,
    Email VARCHAR(30) NOT NULL,
    PRIMARY KEY (Proveedor_ID)
);

-- creando tabla productos
CREATE TABLE Productos(

```

```

    Producto_ID INT NOT NULL,
    Nombre_Producto VARCHAR(50) NOT NULL,
    Proveedor_ID INT NOT NULL,
    Categoria_ID INT NOT NULL,
    Precio FLOAT NOT NULL,
    PRIMARY KEY (PRODUCTO_ID),
    FOREIGN KEY (Categoria_ID) REFERENCES
Categoria (Categoria_ID),
    FOREIGN KEY (Proveedor_ID) REFERENCES
Proveedores(Proveedor_ID)
);

```

-- tabla con los precios

```

CREATE TABLE Precio(
    Precio_ID INT NOT NULL,
    Producto_ID INT NOT NULL,
    Precio_vigencia_inicial FLOAT,
    Precio_vigencia_final FLOAT,
    PRIMARY KEY(Precio_ID),
    FOREIGN KEY(Producto_ID) REFERENCES
Productos(Producto_ID)
);

```

-- tabla con los métodos de pago

```

CREATE TABLE Metodo_Pago (
    Metodo_Pago_ID INT,
    Nombre VARCHAR(50) NOT NULL,
    Descripcion VARCHAR(100),
    PRIMARY KEY(Metodo_Pago_ID)

```



```

);

-- creación de la tabla pedido
CREATE TABLE Pedido(
    Pedido_ID INT NOT NULL,
    Cliente_ID INT NOT NULL,
    Empleado_ID INT NOT NULL,
    Fecha DATE NOT NULL,
    PRIMARY KEY (Pedido_ID),
    FOREIGN KEY (Empleado_ID) REFERENCES Empleados
(Empleado_ID),
    FOREIGN KEY (Cliente_ID) REFERENCES Clientes
(Cliente_ID)
);

-- creando la tabla con los detalles de los
pedidos
CREATE TABLE Detalle_Pedido(
    Detalle_pedido_ID INT NOT NULL,
    Pedido_ID INT NOT NULL,
    Producto_ID INT NOT NULL,
    Cantidad INT NOT NULL,
    Metodo_Pago_ID INT NOT NULL,
    PRIMARY KEY (Detalle_pedido_ID),
    FOREIGN KEY (Pedido_ID) REFERENCES Pedido
(Pedido_ID),
    FOREIGN KEY (Producto_ID) REFERENCES Productos
(Producto_ID),
    FOREIGN KEY (Metodo_Pago_ID) REFERENCES
Metodo_Pago (Metodo_Pago_ID));

```

SCRIPT DE INSERCIÓN DE DATOS

```
USE Emprendimiento_Condimento;
```

```
-- inserción de datos de Categoría
```

```
INSERT INTO CATEGORIA
```

```
VALUES (1,'especia','sustancia vegetal. Suelen ser  
más intensas en sabor y aroma que las hierbas y se  
utilizan en pequeñas cantidades debido a su  
potencia.'),
```

```
(2,'condimento','sustancia de origen vegetal,  
animal o mineral, que se agrega a los alimentos  
para mejorar su sabor o aroma. Los condimentos  
pueden incluir especias e ingredientes como la  
sal, el azúcar, el vinagre, las salsas y los  
aceites.'),
```

```
(3,'hierba','planta comestible o partes de una  
planta que se utilizan para sazonar y dar sabor a  
los alimentos.suelen tener un sabor más suave y  
fresco en comparación con las especias.'),
```

```
(4,'aderezo','conjunto de ingredientes combinados  
que se usan para sazonar la comida y pueden  
convertir un plato soso o sin gracia en un plato  
sabroso.'),
```

```
(5,'condimento liquido','permiten personalizar el  
sabor de tus comidas y utilizarse para marinar,  
aderezar o acompañar tus elaboraciones');
```

```
-- inserción de datos de la tabla Clientes
```

```

INSERT INTO CLIENTES

VALUES (1,'Chino',116574326,'Calle falsa
123','springfield','Buenos
Aires','chino@gmail.com'),

(2,'Mercado Juan',351233453,'Montevideo
275','villa
urquiza','Cordoba','mercadojuan@hotmail.com'),

(3,'Cooperativa obrera',114394700,'Belgrano
45','Bahia blanca','buenos
aires','cooperativa@obrera.coop'),

(4,'Verduleria Montes',29934566422,'Alem
1203','cipoletti','neuquen','montesverdu@outlook.c
om'),

(5,'absurdo',29834531213,'Panama 460','general
roca','rio negro','absurdo@gmail.com'),

(6,'Luis Rodriguez',29834531213,'Mendoza
1258','regina','rio
negro','luisrodriguez@gmail.com'),

(7,'mayorista buffalo',261223457,'san martin
1708','mendoza','mendoza','buffalomayor@mail.com')
;

```

-- inserción de datos de la tabla Empleados

```

INSERT INTO EMPLEADOS

VALUES (31685342,'Rosa','delejos','2015-08-
04',180000.30),

(38559639,'Mario','castañeda','2020-05-
09',165000.60),

(30559112,'Roberto','carlos','2018-12-
01',174000.80),

(40236001,'marcelo','rodriguez','2020-02-
21',158000.10),

```

```
(37567187,'Ivana','matos','2019-11-03',164000.20);
```

```
-- inserción de proveedores
```

```
INSERT INTO PROVEEDORES
```

```
VALUES (1,'picantin',1194856932,'calle real  
321','springfield','buenos  
aires','picantin@yahoo.com.ar'),
```

```
(2,'saborizan',3412812114,'tucuman  
226','rosario','santa fe','ventas@saborizan.com'),
```

```
(3,'alicante',1155691103,'villegas 1056','comodoro  
rivadavia','chubut','ventas@alicante.com'),
```

```
(4,'pizca de sal',2994501249,'artigas  
1590','centenario','neuquen','pizcadesal@outlook.c  
om'),
```

```
(5,'callieri',2984435130,'Líbano 1579','general  
roca','rio negro','ventas@callieri.com.ar'),
```

```
(6,'especias norte',3627953324,'artigas  
2343','resistencia','chaco','norteespecias@ventas.  
com');
```

```
-- inserción de datos de productos
```

```
INSERT INTO PRODUCTOS
```

```
VALUES (1,'oregano',3,1,130),
```

```
(2,'ajo en polvo',4,1,150),
```

```
(3,'pimienta negra',5,2,100),
```

```
(4,'albaca',2,2,90),
```

```
(5,'romero',3,3,110),
```

```
(6,'canela',1,1,190),
```

```
(7,'mostaza',6,4,140),
```

```
(8,'perejil',5,3,90),
```

```
(9,'aceite de sesamo',4,5,120),
```

```
(10,'salsa de barbacoa',5,5,140);
```

```
-- insercion de precios
```

```
INSERT INTO PRECIO
```

```
VALUES (1,2,100,150),
```

```
(2,1,90,130),
```

```
(3,3,75,100),
```

```
(4,4,50,90),
```

```
(5,6,110,190),
```

```
(6,5,70,110),
```

```
(7,7,110,140),
```

```
(8,8,60,90),
```

```
(9,9,90,120),
```

```
(10,10,900,140);
```

```
-- inserción de datos de metodo de pago
```

```
INSERT INTO METODO_PAGO
```

```
VALUES (1,'efectivo','con dinero en mano'),
```

```
(2,'debito','se cobra con tarjeta de debito'),
```

```
(3,'credito','credito con 10% de recargo'),
```

```
(4,'transferencia','transferencia bancaria al  
CBU');
```

```
-- inserción de datos de la tabla pedido
```

```
INSERT INTO PEDIDO
```

```
VALUES (1,2,40236001,'2023-09-01'),
```

```
(2,4,38559639,'2023-10-10'),
```

```
(3,1,38559639,'2022-01-11'),
```

```
(4,3,31685342,'2022-06-30'),  
(5,5,30559112,'2021-12-20'),  
(6,2,37567187,'2023-05-04'),  
(7,6,40236001,'2022-03-13'),  
(8,7,38559639,'2023-08-06');
```

```
-- inserción de datos en la tabla detalle_pedido  
INSERT INTO DETALLE_PEDIDO  
VALUES (1,3,2,10,4),  
(2,4,4,10,4),  
(3,1,5,25,3),  
(4,5,6,25,1),  
(5,2,1,10,2),  
(6,3,7,35,1),  
(7,2,6,30,3),  
(8,8,6,8,4);
```

SCRIPT DE VISTAS

```
-- clientes que más compran  
CREATE OR REPLACE view mejores_clientes  
AS  
SELECT c.nombre_cliente AS  
cliente,Count(p.cliente_id) AS total  
FROM clientes c  
LEFT JOIN pedido p
```

```

ON c.cliente_id = p.cliente_id
GROUP BY c.cliente_id
ORDER BY total DESC;

-- los clientes que pagan con transferencia
CREATE OR REPLACE view pago_transferencia AS
SELECT * FROM detalle_pedido where metodo_pago_id
= 4;

-- los productos con sus detalles
CREATE OR REPLACE view join_producto AS
SELECT c.categoria_id,nombre_categoria,
p.nombre_producto, p.precio from categoria c
INNER JOIN productos p ON c.Categoria_ID =
p.categoria_id;

-- producto mas iva
CREATE OR REPLACE view producto_iva AS
SELECT *, round(precio * 1.21 + precio) AS IVA
FROM productos;

-- top productos mas vendidos
CREATE OR REPLACE view cantidad_vendido AS
SELECT producto_id, (SELECT nombre_producto FROM
productos p WHERE d.producto_id = p.producto_id)
AS nombre,
```

```
SUM(cantidad) AS total_vendido FROM detalle_pedido  
d  
GROUP BY producto_id  
ORDER BY total_vendido DESC;
```

SCRIPT DE FUNCIONES

```
-- función para saber el nombre del cliente  
  
DELIMITER $$  
  
CREATE FUNCTION ObtenerNombreCliente(PedidoID INT)  
RETURNS VARCHAR(50)  
DETERMINISTIC  
BEGIN  
    DECLARE NombreCliente VARCHAR(50);  
  
    SELECT c.Nombre_cliente INTO NombreCliente  
    FROM Clientes c  
    JOIN Pedido p ON c.Cliente_ID = p.Cliente_ID  
    WHERE p.Pedido_ID = PedidoID;  
  
    RETURN NombreCliente;  
END $$
```



```

DELIMITER ;

-- SELECT ObtenerNombreCliente(5) AS
NombreClienteDelPedido;

-- calculo promedio por categoria

DELIMITER $$

CREATE FUNCTION
CalcularPrecioPromedioEnCategoria(categoria_id
INT) RETURNS FLOAT
DETERMINISTIC
BEGIN
    DECLARE precio_promedio FLOAT;

    SELECT AVG(Precio) INTO precio_promedio
    FROM Productos
    WHERE Categoria_ID = categoria_id;

    RETURN precio_promedio;
END $$

DELIMITER ;

-- SELECT CalcularPrecioPromedioEnCategoria(1);

```

SCRIPT DE STORED PROCEDURES

```
-- calculo precio total

DELIMITER $$

CREATE PROCEDURE

    calculo_precio_total (IN precio_unitario DECIMAL(10,2),
                        IN cantidad          INT,
                        IN descuento         DECIMAL(10,2))

BEGIN

    DECLARE precio_total DECIMAL(10,2); SET precio_total =
precio_unitario * cantidad * (1 - descuento); SELECT
precio_total AS 'precio_total';

END $$

DELIMITER;

-- CALL calculo_precio_total (10.00,5,0.1);
```

```
-- agregar un nuevo cliente en la tabla 'clientes'
```

```
DELIMITER $$

CREATE PROCEDURE

    insertarcliente(

                        IN cliente_id      INT,
                        IN nombre_cliente  VARCHAR(100),
                        IN telefono        VARCHAR(15),
                        IN direccion       VARCHAR(100),
                        IN ciudad          VARCHAR(50),
                        IN provincia       VARCHAR(50),
                        IN email           VARCHAR(100)
                    )

BEGIN
```

```
    INSERT INTO clientes
```

```

(
    cliente_id,
    nombre_cliente,
    telefono,
    direccion,
    ciudad,
    provincia,
    email
)
VALUES
(
    cliente_id,
    nombre_cliente,
    telefono,
    direccion,
    ciudad,
    provincia,
    email
);

END $$

DELIMITER;

-- CALL InsertarCliente(100, 'Nombre del Cliente',
'123456789', 'Av. Roca 1233', 'Ciudad', 'Provincia',
'cliente@mail.com');

-- eliminar un cliente de la tabla 'clientes'
DELIMITER $$

CREATE PROCEDURE
    eliminarcliente (IN cliente_id INT)
BEGIN

```

```

DELETE

FROM    clientes

WHERE   cliente_id = cliente_id;

END $$

DELIMITER ;

-- CALL EliminarCliente(1);

```

SCRIPT DE TRIGGERS

```

-- trigger con registro de inserción de datos
relacionado con productos y precios

```

```

DROP TRIGGER IF EXISTS precio_nuevo;

DELIMITER $$

CREATE TRIGGER precio_nuevo
BEFORE INSERT ON precio
FOR EACH ROW
BEGIN

    INSERT INTO Registro_Cambios_Precios
    (Producto_ID, Precio_Anterior)

    VALUES (NEW.Producto_ID,
NEW.Precio_vigencia_inicial);

END;

$$

```

```
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS precio_eliminado;
```

```
DELIMITER $$
```

```
CREATE TRIGGER precio_eliminado
```

```
AFTER DELETE ON Precio
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO Registro_Precios_Eliminados  
    (Producto_ID, Precio_Eliminado)
```

```
    VALUES (OLD.Producto_ID,  
    OLD.Precio_vigencia_inicial);
```

```
END; $$
```

```
DELIMITER ;
```

```
-- trigger de registro de productos
```

```
DROP TABLE IF EXISTS registro_producto;
```

```
CREATE TABLE registro_producto (
```

```
    accion_realizada VARCHAR(10) NOT NULL,
```

```
    producto_id INT NOT NULL,
```

```
    nombre_producto VARCHAR(50),
```

```
    proveedor_id INT,
```

```
    categoria_id INT,
```

```

        precio FLOAT,

        fecha TIMESTAMP

    );

-- Crear un trigger after para insertar registros
en la tabla de registro

DROP TRIGGER IF EXISTS log_productos;

DELIMITER //

CREATE TRIGGER log_productos
AFTER INSERT ON productos
FOR EACH ROW
BEGIN

    INSERT INTO registro_productos
    (accion_realizada, producto_id, nombre_producto,
    proveedor_id, categoria_id, precio, fecha)

    VALUES ('INSERT', NEW.producto_id,
    NEW.nombre_producto, NEW.proveedor_id,
    NEW.categoria_id, NEW.precio, NOW());

END;

//

DELIMITER ;

```