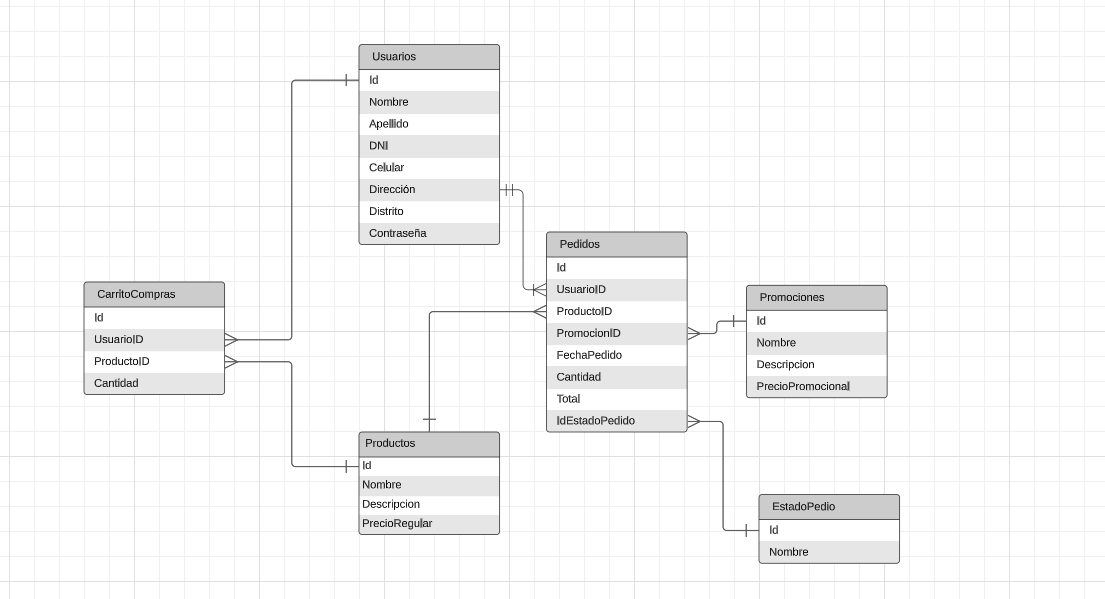
**DOCUMENTO DE DISEÑO DE LA BASE DE DATOS**

1. **Objetivos**

* Ofrecer una plataforma de compra en línea: El objetivo principal es proporcionar a los usuarios una plataforma intuitiva y segura donde puedan realizar compras de pollos a la brasa de forma conveniente desde la comodidad de sus hogares.
* Facilitar la experiencia de compra: El sitio web debe garantizar una experiencia de compra fluida y amigable para los usuarios, desde la navegación por los productos hasta el proceso de pago. Esto implica la implementación de un diseño intuitivo, un carrito de compras funcional y opciones de búsqueda y filtrado eficientes.
* Mostrar información detallada de los productos: Es importante que los usuarios tengan acceso a información completa y precisa sobre los pollos a la brasa disponibles. Esto incluye detalles como el nombre del producto, descripción, precio regular y posibles promociones aplicables.
* Brindar promociones y descuentos atractivos: El sitio web debe permitir la creación y gestión de promociones especiales, como descuentos por tiempo limitado o combos especiales, para atraer a los clientes y aumentar las ventas.
* Gestionar cuentas de usuarios: El objetivo es proporcionar a los usuarios la posibilidad de crear una cuenta personalizada en el sitio web, donde puedan almacenar sus datos personales, realizar un seguimiento de sus pedidos y recibir actualizaciones sobre promociones y novedades.
* Administrar pedidos y seguimiento: El sistema debe permitir la gestión eficiente de los pedidos realizados, incluyendo la asignación de números de seguimiento para que los usuarios puedan rastrear el estado de sus entregas.
* Garantizar la seguridad de los datos: Se debe implementar una sólida seguridad en la base de datos para proteger la información personal y financiera de los usuarios, utilizando técnicas de encriptación y buenas prácticas de seguridad.
* Optimizar el rendimiento y la escalabilidad: El sitio web debe estar diseñado y desarrollado de manera eficiente, garantizando un rendimiento rápido y confiable incluso en momentos de alta demanda. Además, debe ser escalable para adaptarse al crecimiento futuro del negocio.

1. **Diagrama de entidad-relación (DER):**

****

1. **Descripción de tablas y atributos:**
2. Tabla "Usuarios"

ID (int, autoincremental, clave primaria): Identificador único del usuario.

Nombre (varchar(50)): Nombre del usuario.

Apellido (varchar(50)): Apellido del usuario.

DNI (varchar(8)): Número de identificación personal del usuario.

Celular (varchar(15)): Número de teléfono celular del usuario.

Direccion (varchar(100)): Dirección del usuario.

Distrito (varchar(50)): Distrito en el que reside el usuario.

Contraseña (varchar(50)): Contraseña del usuario.

1. Tabla "Productos"

ID (int, autoincremental, clave primaria): Identificador único del producto.

Nombre (varchar(100)): Nombre del producto.

Descripcion (varchar(255)): Descripción del producto.

PrecioRegular (decimal(10,2)): Precio regular del producto.

1. Tabla "Promociones"

ID (int, autoincremental, clave primaria): Identificador único de la promoción.

Nombre (varchar(100)): Nombre de la promoción.

Descripcion (varchar(255)): Descripción de la promoción.

PrecioPromocional (decimal(10,2)): Precio promocional del producto.

1. Tabla "Pedidos"

ID (int, autoincremental, clave primaria): Identificador único del pedido.

UsuarioID (int, clave foránea a la tabla "Usuarios"): ID del usuario que realizó el pedido.

ProductoID (int, clave foránea a la tabla "Productos"): ID del producto solicitado.

PromocionID (int, clave foránea a la tabla "Promociones", opcional): ID de la promoción aplicada al pedido (si existe).

FechaPedido (datetime): Fecha y hora en que se realizó el pedido.

Cantidad (int): Cantidad solicitada del producto.

Total (decimal(10,2)): Total del pedido.

1. Tabla "CarritoDeCompras":

CarritoID (int, autoincremental, clave primaria): Identificador único del carrito.

UsuarioID (int, clave foránea a la tabla "Usuarios"): ID del usuario al que pertenece el carrito.

ProductoID (int, clave foránea a la tabla "Productos"): ID del producto agregado al carrito.

Cantidad (int): Cantidad del producto agregado al carrito.

1. Tabla "EstadoPedido":

EstadoID (int, autoincremental, clave primaria): Identificador único del estado de pedido.

NombreEstado (varchar(50)): Nombre del estado del pedido (por ejemplo, "En proceso", "Enviado", "Entregado").

1. Tabla "DetallePedido":

DetalleID (int, autoincremental, clave primaria): Identificador único del detalle del pedido.

PedidoID (int, clave foránea a la tabla "Pedidos"): ID del pedido al que pertenece el detalle.

ProductoID (int, clave foránea a la tabla "Productos"): ID del producto incluido en el detalle del pedido.

Cantidad (int): Cantidad del producto en el detalle del pedido.

PrecioUnitario (decimal(10,2)): Precio unitario del producto en el detalle del pedido.

1. Tabla "MetodosPago":

MetodoID (int, autoincremental, clave primaria): Identificador único del método de pago.

NombreMetodo (varchar(50)): Nombre del método de pago (por ejemplo, "Tarjeta de crédito", "Pago en efectivo", etc.).

1. **Scripts SQL:**

Aquí tienes los scripts SQL para crear las tablas en una base de datos MySQL:

1. Tabla Usuarios

CREATE TABLE Usuarios (

ID INT AUTO\_INCREMENT PRIMARY KEY,

Nombre VARCHAR(50),

Apellido VARCHAR(50),

DNI VARCHAR(8),

Celular VARCHAR(15),

Direccion VARCHAR(100),

Distrito VARCHAR(50),

Contraseña VARCHAR(50)

);

1. Tabla Productos

CREATE TABLE Productos (

ID INT AUTO\_INCREMENT PRIMARY KEY,

Nombre VARCHAR(100),

Descripcion VARCHAR(255),

PrecioRegular DECIMAL(10,2)

);

1. Tabla Promociones

CREATE TABLE Promociones (

ID INT AUTO\_INCREMENT PRIMARY KEY,

Nombre VARCHAR(100),

Descripcion VARCHAR(255),

PrecioPromocional DECIMAL(10,2)

);

1. Tabla Pedidos

CREATE TABLE Pedidos (

ID INT AUTO\_INCREMENT PRIMARY KEY,

UsuarioID INT,

ProductoID INT,

PromocionID INT,

FechaPedido DATETIME,

Cantidad INT,

Total DECIMAL(10,2),

FOREIGN KEY (UsuarioID) REFERENCES Usuarios(ID),

FOREIGN KEY (ProductoID) REFERENCES Productos(ID),

FOREIGN KEY (PromocionID) REFERENCES Promociones(ID)

);

1. **Consultas generales utilizadas**
2. Obtener todos los usuarios:

SELECT \* FROM Usuarios;

1. Obtener todos los productos:

SELECT \* FROM Productos;

1. Obtener todos los pedidos realizados:

SELECT \* FROM Pedidos;

1. Obtener el detalle de un pedido específico con información del usuario, producto y promoción (si existe):

SELECT Pedidos.ID, Usuarios.Nombre, Usuarios.Apellido, Productos.Nombre, Promociones.Nombre, Pedidos.FechaPedido, Pedidos.Cantidad, Pedidos.Total

FROM Pedidos

INNER JOIN Usuarios ON Pedidos.UsuarioID = Usuarios.ID

INNER JOIN Productos ON Pedidos.ProductoID = Productos.ID

LEFT JOIN Promociones ON Pedidos.PromocionID = Promociones.ID

WHERE Pedidos.ID = <ID\_DEL\_PEDIDO>;