

Universidad Peruana Los Andes

Ingeniería de Sistemas y Computación

Huancayo Tec – Tienda Online

Curso: Base de Datos II

Docente: Raul Enrique Fernandez Bejarano

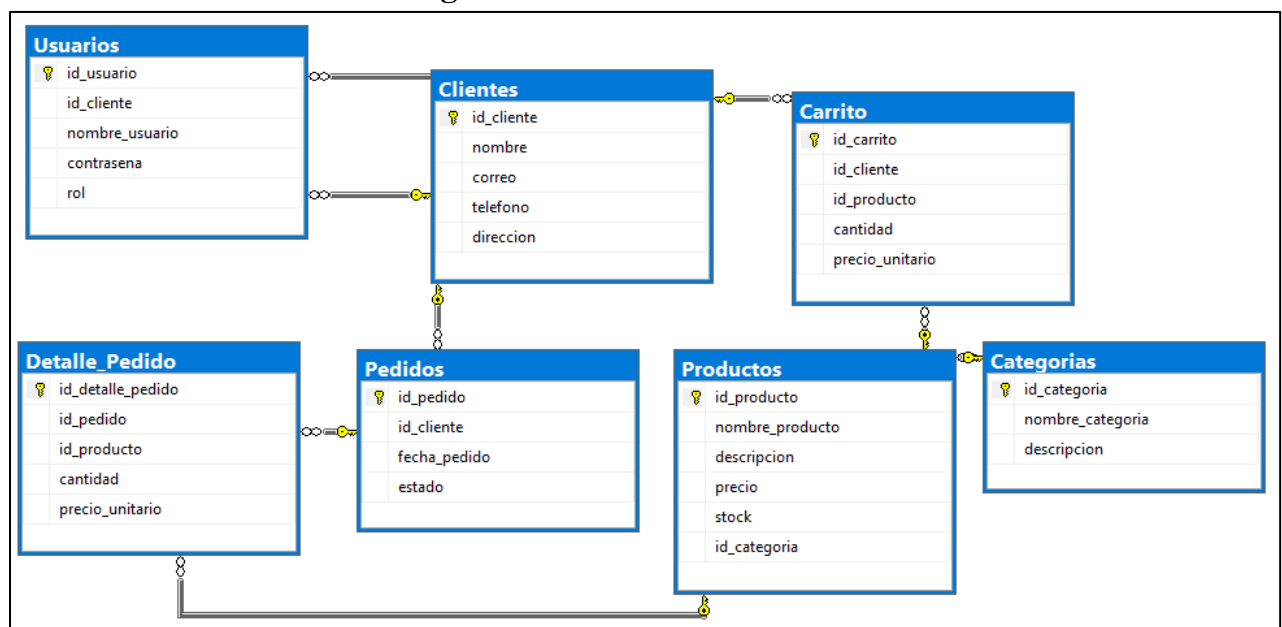
Alumno: Mosquera Zevallos Valerio

Resumen

El sistema de "Tienda Online" está diseñado para gestionar usuarios, clientes, empleados, productos y pedidos dentro de una tienda en línea. La aplicación incluye funcionalidades para el manejo de un carrito de compras, la gestión de inventarios y la generación de pedidos, proporcionando una experiencia completa para administradores y usuarios finales.

El sistema consta de una interfaz web implementada en Jakarta EE, que se conecta a una base de datos SQL Server para almacenar y gestionar la información.

Diagrama de la Base de Datos



Documentación del Sistema de Tienda Online

Introducción

El sistema desarrollado permite gestionar una tienda online, incluyendo el manejo de usuarios, clientes, empleados, productos, pedidos y carrito de compras. Está diseñado para proporcionar una experiencia de usuario eficiente y flexible, con un backend robusto basado en una base de datos SQL bien estructurada.

Descripción General del Sistema

El sistema consta de las siguientes características principales:

1. **Gestor de Usuarios:**
 - Los usuarios pueden ser administradores, empleados o clientes.
 - Los datos se almacenan en una tabla llamada Usuarios.
2. **Gestor de Productos:**
 - Los productos disponibles se registran con sus atributos (nombre, descripción, precio y stock) en la tabla Productos.
3. **Carrito de Compras:**
 - Permite a los usuarios agregar productos y gestionar la cantidad antes de realizar pedidos.
4. **Sistema de Pedidos:**
 - Los pedidos incluyen un detalle de los productos seleccionados y un total calculado.
5. **Base de Datos con Relaciones:**
 - Tablas normalizadas con claves foráneas para garantizar integridad referencial.

Vistas de la Aplicación

Login

Iniciar Sesión

Usuario:

Contraseña:

Iniciar Sesión

¿No tienes cuenta? [Regístrate aquí](#)

Credenciales incorrectas.

Iniciar Sesión

Usuario:

Contraseña:

Iniciar Sesión

¿No tienes cuenta? [Regístrate aquí](#)

Registro

Registro

Nombre:

Apellido:

Usuario:

Contraseña:

Correo:

Teléfono:

Registrar

Página Principal del Usuario


HuancayoTec

Productos Disponibles

Nombre	Descripción	Precio	Acciones
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99	Agregar al Carrito
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99	Agregar al Carrito
Zapatillas Running	Zapatillas para correr, color negro	S/. 59.99	Agregar al Carrito
Mochila Escolar	Mochila con varios compartimentos, 25L	S/. 29.99	Agregar al Carrito
Reloj Inteligente	Reloj inteligente con múltiples funciones de salud	S/. 199.99	Agregar al Carrito
Auriculares Bluetooth	Auriculares inalámbricos con cancelación de ruido	S/. 89.99	Agregar al Carrito

[Ver Carrito](#)
[Cerrar sesión](#)

Pagina para ver el Carrito


HuancayoTec

Carrito de Compras

Nombre	Descripción	Precio
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99

[Regresar a productos](#)

Página Principal para el Empleado

**HuancayoTec**

Bienvenido, Empleado

[Ver Clientes](#) | [Ver Pedidos](#)

Cerrar sesión

Pagina para ver clientes

**HuancayoTec**

Lista de Clientes

ID Cliente	Nombre	Apellido	Correo	Teléfono

Agregar Cliente

Pagina para agregar Clientes

**HuancayoTec**

Agregar Cliente

ID Usuario:

Nombre:

Apellido:

Correo:

Teléfono:

Agregar Cliente

Volver a la lista

Estructura de la Aplicación en Apache NetBeans

Semana 16-1.0-SNAPSHOT [0000000...]

```

├── Web Pages
│   ├── META-INF
│   ├── WEB-INF
│   │   └── beans.xml      # Archivo de configuración para CDI (Context and Dependency
Injection).
│   ├── web.xml          # Archivo de configuración principal para el despliegue en el servidor.
│   └── recursos
│       ├── estilo.css    # Estilos generales para la aplicación web.
│       ├── logoUPLA.png  # Imagen del logo utilizada en las páginas.
│       ├── script.js     # Scripts de funcionalidad para la interfaz.
│       └── tablas.css     # Estilos específicos para las tablas de datos.
├── agregar_cliente.jsp   # Formulario para agregar nuevos clientes.
├── carrito.jsp           # Página para gestionar el carrito de compras.
├── clientes.jsp          # Página para mostrar y gestionar clientes.
├── home.jsp              # Página principal para usuarios comunes.
├── home_empleado.jsp     # Página principal para empleados.
├── login.jsp             # Página de inicio de sesión.
├── pedidos.jsp           # Página para gestionar pedidos.
├── registro.jsp          # Página para registrar nuevos usuarios.
└── usuarios.jsp          # Página para mostrar y gestionar usuarios.

```

```

├── Source Packages
│   ├── com.mycompany.semana16
│   │   ├── resources      # Recursos auxiliares de configuración o utilidades.
│   │   └── com.tiendaonline.controller
│   │       ├── CarritoController.java # Controlador para gestionar la lógica del carrito.
│   │       ├── ClienteController.java # Controlador para operaciones relacionadas con clientes.
│   │       ├── EmpleadoController.java # Controlador para gestionar empleados.
│   │       ├── HomeController.java    # Controlador para navegación y vistas principales.
│   │       └── UsuarioController.java # Controlador para gestionar usuarios.
│   │   └── com.tiendaonline.model
│   │       ├── Carrito.java          # Modelo de datos para el carrito.
│   │       ├── Cliente.java          # Modelo de datos para clientes.
│   │       ├── Pedido.java           # Modelo de datos para pedidos.
│   │       ├── Producto.java         # Modelo de datos para productos.
│   │       └── Usuario.java          # Modelo de datos para usuarios.
│   │   └── com.tiendaonline.service
│   │       ├── CarritoService.java   # Lógica de negocio para el carrito.
│   │       ├── ClienteService.java   # Lógica de negocio para clientes.
│   │       ├── EmpleadoService.java  # Lógica de negocio para empleados.
│   │       ├── ProductoService.java  # Lógica de negocio para productos.
│   │       └── UsuarioService.java    # Lógica de negocio para usuarios.
│   │   └── com.tiendaonline.util
│   │       ├── DBConnection.java     # Clase para gestionar la conexión a la base de datos.
│   │       └── TestConnectionServlet.java # Servlet para probar la conexión a la base de datos.
└── Project Files

```

| |— pom.xml # Archivo de configuración de Maven para la gestión de dependencias y compilación.
 | |— nb-configuration.xml # Archivo de configuración específico del IDE NetBeans.

Base de Datos

Estructura de la Base de Datos

La base de datos se denomina **TiendaOnline** y consta de las siguientes tablas principales:

1. **Usuarios:**
 - Almacena información de los usuarios registrados.
 - Campos importantes: id_usuario, nombre_usuario, contrasena, rol.
2. **Clientes:**
 - Relaciona un usuario con el rol de cliente.
 - Incluye información adicional como fecha_registro.
3. **Empleado:**
 - Relaciona un usuario con el rol de empleado.
 - Atributos adicionales: puesto.
4. **Productos:**
 - Almacena los productos disponibles en la tienda.
 - Campos importantes: id_producto, nombre_producto, precio, stock.
5. **Pedidos:**
 - Representa los pedidos realizados por los usuarios.
 - Incluye detalles como estado y total.
6. **DetallePedido:**
 - Contiene el desglose de los productos en cada pedido.
7. **Carrito:**
 - Administra los productos agregados por los usuarios antes de confirmar el pedido.

Relación entre Tablas

La base de datos está diseñada con relaciones para garantizar la integridad de los datos:

- **Usuarios y Clientes/Empleado:** Los clientes y empleados se relacionan con la tabla Usuarios mediante la clave foránea id_usuario.
- **Pedidos y DetallePedido:** Cada pedido tiene varios detalles, asociados mediante la clave foránea id_pedido.
- **Carrito y Productos:** Los productos seleccionados por un usuario se relacionan con Productos y Usuarios.

Scripts SQL

La base de datos incluye los siguientes scripts:

Creación de Tablas

Se definen todas las tablas necesarias, asegurando integridad referencial mediante claves foráneas.

Procedimientos Almacenados

- **BackupDatabase:** Realiza un respaldo de la base de datos en una ruta específica.
- **RestoreDatabase:** Restaura la base de datos desde un archivo de respaldo.

Inserción de Datos

- Ejemplo de datos precargados para usuarios, productos y carritos.

Detalles de Implementación

Frontend

- Interfaz de usuario sencilla, basada en JSP.
- Ejemplo: clientes.jsp muestra una tabla con clientes registrados y un botón para agregar nuevos clientes.

Backend

- Controladores gestionan las acciones solicitadas desde el frontend.
- Funciones implementadas en Java para procesar lógica de negocio.

Seguridad

- Contraseñas encriptadas antes de almacenarse en la base de datos.
- Validación de roles para restringir accesos.

Respaldo y Recuperación de la Base de Datos

Procedimientos para Backup

1. Especifique la ruta de almacenamiento para el archivo de respaldo.
2. Ejecute EXEC BackupDatabase @BackupPath = 'C:\Respaldos';.

Procedimientos para Restauración

1. Asegúrese de tener un archivo de respaldo válido.
2. Ejecute EXEC RestoreDatabase @BackupPath = 'C:\Respaldos\TiendaOnline_YYYY-MM-DD_HH-MM-SS.bak';.

Script de mi base de datos

```
-- Crear la base de datos
CREATE DATABASE TiendaOnline;
GO

USE TiendaOnline;
GO

-- Crear tablas
CREATE TABLE Usuarios (
    id_usuario INT PRIMARY KEY IDENTITY(1,1),
    nombre NVARCHAR(50) NOT NULL,
    apellido NVARCHAR(50) NOT NULL,
    nombre_usuario NVARCHAR(50) NOT NULL UNIQUE,
    contraseña NVARCHAR(100) NOT NULL,
    correo NVARCHAR(100),
    telefono NVARCHAR(15),
    rol NVARCHAR(20) DEFAULT 'cliente'
);
```



```
CREATE TABLE Clientes (
    id_cliente INT PRIMARY KEY IDENTITY(1,1),
    id_usuario INT NOT NULL,
    fecha_registro DATETIME DEFAULT GETDATE(),
    CONSTRAINT FK_Cliente_Usuario FOREIGN KEY (id_usuario) REFERENCES
    Usuarios(id_usuario)
);
```

```
CREATE TABLE Empleado (
    id_empleado INT PRIMARY KEY IDENTITY(1,1),
    id_usuario INT NOT NULL,
    puesto NVARCHAR(100),
    CONSTRAINT FK_Empleado_Usuario FOREIGN KEY (id_usuario) REFERENCES
    Usuarios(id_usuario)
);
```

```
CREATE TABLE Categorías (
    id_categoria INT IDENTITY(1,1) PRIMARY KEY,
    nombre_categoria NVARCHAR(100) NOT NULL,
    descripcion NVARCHAR(255)
);
```

```
CREATE TABLE Productos (
    id_producto INT PRIMARY KEY IDENTITY(1,1),
    nombre_producto NVARCHAR(100) NOT NULL,
    descripcion NVARCHAR(255),
    precio DECIMAL(10, 2) NOT NULL,
    stock INT NOT NULL
);
```

```
CREATE TABLE Pedidos (
    id_pedido INT PRIMARY KEY IDENTITY(1,1),
    id_usuario INT NOT NULL,
    fecha_pedido DATETIME DEFAULT GETDATE(),
    estado NVARCHAR(20) DEFAULT 'Pendiente',
    total DECIMAL(10, 2) NOT NULL,
    CONSTRAINT FK_Pedidos_Usuario FOREIGN KEY (id_usuario) REFERENCES
    Usuarios(id_usuario)
);
```

```
CREATE TABLE DetallePedido (
    id_detalle INT PRIMARY KEY IDENTITY(1,1),
    id_pedido INT NOT NULL,
    id_producto INT NOT NULL,
    cantidad INT NOT NULL,
    subtotal DECIMAL(10, 2) NOT NULL,
    CONSTRAINT FK_DetallePedido_Pedido FOREIGN KEY (id_pedido) REFERENCES
    Pedidos(id_pedido),
```

```
CONSTRAINT FK_DetallePedido_Producto FOREIGN KEY (id_producto) REFERENCES
Productos(id_producto)
);
```

```
CREATE TABLE Carrito (
    id_carrito INT PRIMARY KEY IDENTITY(1,1),
    id_usuario INT NOT NULL,
    id_producto INT NOT NULL,
    cantidad INT NOT NULL DEFAULT 1,
    CONSTRAINT FK_Carrito_Usuario FOREIGN KEY (id_usuario) REFERENCES
Usuarios(id_usuario),
    CONSTRAINT FK_Carrito_Producto FOREIGN KEY (id_producto) REFERENCES
Productos(id_producto)
);
```

```
-- Procedimientos almacenados para backup y restauración
-- Backup de la base de datos
```

```
CREATE PROCEDURE BackupDatabase
    @BackupPath NVARCHAR(MAX)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @BackupFile NVARCHAR(MAX) = @BackupPath + '\TiendaOnline_' +
CONVERT(NVARCHAR(20), GETDATE(), 120) + '.bak';
    BACKUP DATABASE TiendaOnline TO DISK = @BackupFile
    WITH FORMAT, INIT, NAME = 'TiendaOnline Backup';
    PRINT 'Backup completado exitosamente: ' + @BackupFile;
END;
GO
```

```
-- Restauración de la base de datos
```

```
CREATE PROCEDURE RestoreDatabase
    @BackupPath NVARCHAR(MAX)
AS
BEGIN
    SET NOCOUNT ON;

    RESTORE DATABASE TiendaOnline
    FROM DISK = @BackupPath
    WITH REPLACE, RECOVERY;
    PRINT 'Restauración completada exitosamente desde: ' + @BackupPath;
END;
GO
```

```
-- Inserción en la tabla Carrito
```

```
INSERT INTO Carrito (id_usuario, id_producto, cantidad)
VALUES
```

(1, 2, 3), -- Usuario 1 agregó 3 unidades del producto 2 (Pantalón de Jeans)

(2, 4, 1); -- Usuario 2 agregó 1 unidad del producto 4 (Mochila Escolar)

-- Inserción en la tabla Usuarios

INSERT INTO Usuarios (nombre, apellido, nombre_usuario, contrasena, correo, telefono)

('VALERIO', 'MOSQUERA ZEVALLOS', 'valerio_mosquera', 'contrasena123', 'valerio@email.com', '999999999'),

('ENDI WILMER', 'ALFARO BARRIENTOS', 'endi_wilmer', 'contrasena123', 'endi@email.com', '999999999'),

('JHON WILLIAM', 'QUISPE GAMBOA', 'jhon_william', 'contrasena123', 'jhon@email.com', '999999999'),

('ELVIS', 'RAMIREZ ORE', 'elvis_ramirez', 'contrasena123', 'elvis@email.com', '999999999'),

('ANGELO PERCY', 'ACUINO ALEJO', 'angelo_percy', 'contrasena123', 'angelo@email.com', '999999999'),

('ANDRE', 'BUENDIA CHIPANA', 'andre_buendia', 'contrasena123', 'andre@email.com', '999999999'),

('DENISSE ALEXANDRA', 'MOORI PIÑA', 'denisse_alexandra', 'contrasena123', 'denisse@email.com', '999999999'),

('CARLOS ELMER', 'ROSALES RODRIGUEZ', 'carlos_elmer', 'contrasena123', 'carlos@email.com', '999999999'),

('FLOR XIOMARA', 'MEDINA SALAZAR', 'flor_xiomara', 'contrasena123', 'flor@email.com', '999999999'),

('LEONARDO DANY', 'AVELINO URIBE', 'leonardo_dany', 'contrasena123', 'leonardo@email.com', '999999999'),

('LEONARDO HERIBERTO', 'VILLAVICENCIO REYES', 'leonardo_heriberto', 'contrasena123', 'leonardo2@email.com', '999999999'),

('LUIS ANGEL GONZALO', 'AGUILERA', 'luis_angel', 'contrasena123', 'luis@email.com', '999999999'),

('BRAYAN', 'APOMAYTA CUBA', 'brayan_apomayta', 'contrasena123', 'brayan@email.com', '999999999'),

('WILSON SAMIR', 'FLORES ESTRELLA', 'wilson_samir', 'contrasena123', 'wilson@email.com', '999999999'),

('VICTOR JESUS', 'ARIAS LUYO', 'victor_jesus', 'contrasena123', 'victor@email.com', '999999999'),

('GIANFRANCO SERGIO', 'VILLOSLADA VILCA', 'gianfranco_sergio', 'contrasena123', 'gianfranco@email.com', '999999999'),

('RICHARD ALEJANDRO', 'GAMBOA FERNANDEZ', 'richard_alejandro', 'contrasena123', 'richard@email.com', '999999999'),

('ROGGER JOSUE', 'PALMA BALDEON', 'rogger_josue', 'contrasena123', 'rogger@email.com', '999999999'),

('FABRICIO CESAR', 'ROSALES ALANYA', 'fabricio_cesar', 'contrasena123', 'fabricio@email.com', '999999999'),

('SAMIR SABDIEL', 'ROJAS TAMARA', 'samir_sabdiel', 'contrasena123', 'samir@email.com', '999999999'),

('DIEGO ARMANDO', 'MULATO CRUZ', 'diego_armando', 'contrasena123', 'diego@email.com', '999999999'),

('OLIVER', 'ESPINOZA FERNANDEZ', 'oliver_espinoza', 'contrasena123', 'oliver@email.com', '999999999'),

```
('LUIS ANGEL', 'CHAVEZ MAGUIÑA', 'luis_angel_chavez', 'contrasena123',  
'luis_chavez@email.com', '999999999'),  
( 'ANTONY JHEANPIER', 'MAYTA POMA', 'antony_jheanpier', 'contrasena123',  
'antony@email.com', '999999999'),  
( 'DAVID AARON', 'PAPUICO RAFAEL', 'david_aaron', 'contrasena123', 'david@email.com',  
'999999999'),  
( 'JESUS MIGUEL', 'EGOAVIL MACHADO', 'jesus_miguel', 'contrasena123', 'jesus@email.com',  
'999999999'),  
( 'JHON', 'SMITH ROMAN ROQUE', 'jhon_smith', 'contrasena123', 'jhon_smith@email.com',  
'999999999'),  
( 'JOEL KEVIN', 'SORIANO TIMOTEO', 'joel_kevin', 'contrasena123', 'joel@email.com',  
'999999999'),  
( 'THOMAS', 'TIRACCAYA CASTILLON', 'thomas_tiraccaya', 'contrasena123', 'thomas@email.com',  
'999999999'),  
( 'YURIAN RODRIGO', 'QUISPE VICTORIA', 'yurian_rodrigo', 'contrasena123',  
'yurian@email.com', '999999999'),  
( 'EVELYN MILAGROS', 'LIMA MARTINEZ', 'evelyn_milagros', 'contrasena123',  
'evelyn@email.com', '999999999'),  
( 'EMMANUEL GADDIEL', 'GUADALUPE CARBAJAL', 'emmanuel_gaddiel', 'contrasena123',  
'emmanuel@email.com', '999999999');
```

-- Inserción en la tabla Productos

```
INSERT INTO Productos (nombre_producto, descripcion, precio, stock)
```

```
VALUES
```

```
('Camiseta Deportivo', 'Camiseta deportiva de algodón, talla M', 19.99, 100),  
( 'Pantalón de Jeans', 'Pantalón de jeans de corte moderno, talla 32', 39.99, 50),  
( 'Zapatillas Running', 'Zapatillas para correr, color negro', 59.99, 150),  
( 'Mochila Escolar', 'Mochila con varios compartimentos, 25L', 29.99, 80),  
( 'Reloj Inteligente', 'Reloj inteligente con múltiples funciones de salud', 199.99, 30),  
( 'Auriculares Bluetooth', 'Auriculares inalámbricos con cancelación de ruido', 89.99, 120),  
( 'Chaqueta Impermeable', 'Chaqueta impermeable y transpirable, talla L', 79.99, 40),  
( 'Gorra Deportiva', 'Gorra ajustable con diseño deportivo, color gris', 14.99, 200),  
( 'Cargador Solar', 'Cargador portátil solar, 10000 mAh', 34.99, 60),  
( 'Silla Gamer', 'Silla ergonómica para gamers, con reposabrazos ajustables', 149.99, 25),  
( 'Lámpara LED', 'Lámpara LED de escritorio con ajuste de intensidad, color blanco', 24.99, 90),  
( 'Bocina Bluetooth', 'Bocina portátil Bluetooth con sonido estéreo, color rojo', 49.99, 75),  
( 'Vajilla de Cerámica', 'Juego de vajilla de cerámica de 12 piezas, diseño moderno', 59.99, 50),  
( 'Taza Termo', 'Taza termo de acero inoxidable, mantiene la temperatura, 500ml', 19.99, 120),  
( 'Kit de Maquillaje', 'Kit completo de maquillaje con productos para rostro, ojos y labios', 69.99,  
40),  
( 'Bicicleta de Montaña', 'Bicicleta de montaña, 21 velocidades, frenos de disco', 399.99, 15);
```

-- Inserción del primer empleado

```
INSERT INTO Usuarios (nombre, apellido, nombre_usuario, contrasena, correo, telefono, rol)
```

```
VALUES ('Ivan', 'Aimituma', 'andy', '123', 'andy@empresa.com', '987654321', 'empleado');
```

-- Inserción del segundo empleado

```
INSERT INTO Usuarios (nombre, apellido, nombre_usuario, contrasena, correo, telefono, rol)
```

```
VALUES ('Luis', 'Aquino', 'lucho', '123', 'luchito@empresa.com', '912345678', 'empleado');
```

```
-- Insertar registros con fecha personalizada
```

```
INSERT INTO Clientes (id_usuario, fecha_registro)
```

```
VALUES (1, '2024-12-19 10:00:00');
```

```
INSERT INTO Clientes (id_usuario, fecha_registro)
```

```
VALUES (2, '2024-12-19 11:00:00');
```

```
INSERT INTO Clientes (id_usuario, fecha_registro)
```

```
VALUES (3, '2024-12-19 12:00:00');
```

```
INSERT INTO Clientes (id_usuario, fecha_registro)
```

```
VALUES (4, '2024-12-19 13:00:00');
```

```
INSERT INTO Clientes (id_usuario, fecha_registro)
```

```
VALUES (5, '2024-12-19 14:00:00');
```

```
-- Ver los datos en cada tabla antes de eliminarlas
```

```
SELECT * FROM Usuarios;
```

```
SELECT * FROM Clientes;
```

```
SELECT * FROM Empleado;
```

```
SELECT * FROM Categorías;
```

```
SELECT * FROM Productos;
```

```
SELECT * FROM Pedidos;
```

```
SELECT * FROM DetallePedido;
```

```
SELECT * FROM Carrito;
```

```
-- Eliminar las tablas en orden adecuado para evitar problemas con las claves foráneas
```

```
DROP TABLE IF EXISTS Carrito;
```

```
DROP TABLE IF EXISTS DetallePedido;
```

```
DROP TABLE IF EXISTS Pedidos;
```

```
DROP TABLE IF EXISTS Productos;
```

```
DROP TABLE IF EXISTS Categorías;
```

```
DROP TABLE IF EXISTS Empleado;
```

```
DROP TABLE IF EXISTS Clientes;
```

```
DROP TABLE IF EXISTS Usuarios;
```