**PROYECTO GRUPAL**

“Desarrollo de una plataforma digital para el control externo y interno de ACHI VALENTINAS QUINTA RESTAURANT”

**GRUPO**:

“INFINITYX”

**DOCENTE**:

ING. HUGO ESPETIA HUAMANGA

NRC:

**ESTUDIANTES:**

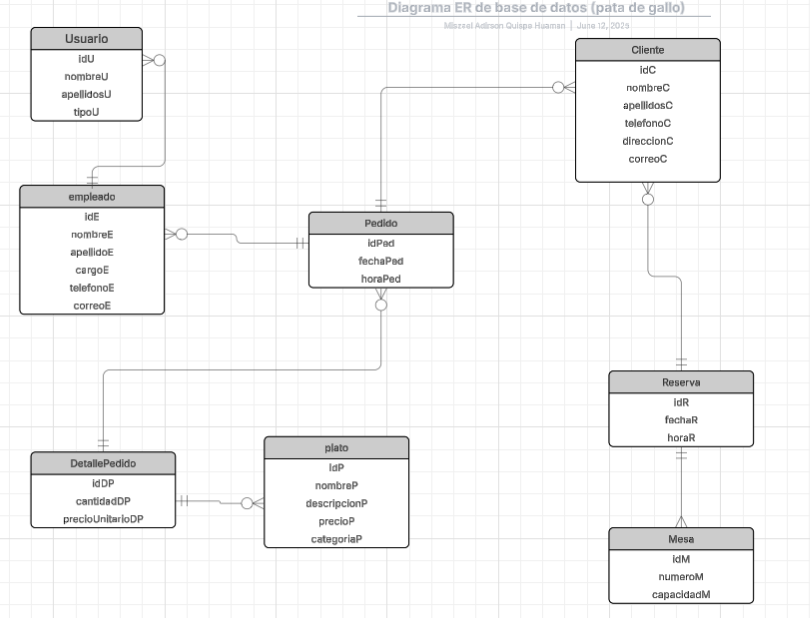
* Quispe Huaman Miszael Adirson
* Zevallos Serrano Brom Jhirdel
* Guillermo Ccahua Laurel

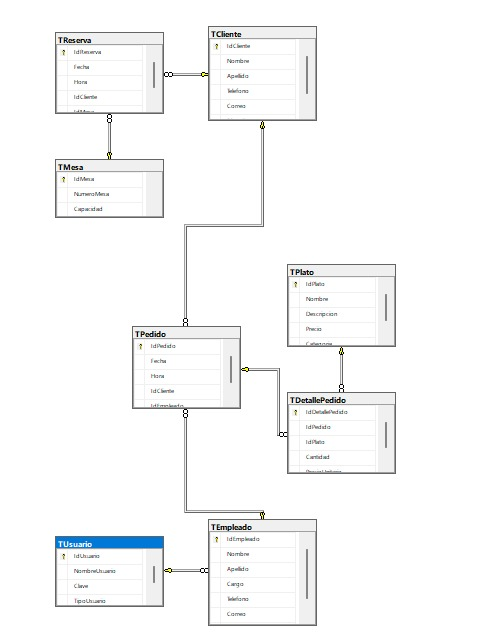
**INDICE**

1. Introducción
2. Diseño de la Base de Datos
   1. Diagrama Entidad-Relación Lógico
   2. Diagrama Entidad-Relación Físico
   3. Diccionario de Datos
   4. Justificación del Diseño
3. Implementación de la Base de Datos
   1. Esquema General de la Base de Datos
   2. Creación de Tablas y Restricciones
   3. Procedimientos Almacenados CRUD
   4. Datos de Prueba
4. Ejemplos de Uso de Procedimientos Almacenados
5. Anexos
   1. Scripts SQL
   2. Acceso al Repositorio GitHub
   3. Capturas o Evidencias de Funcionamiento
6. Presentación del Proyecto
   1. Descripción General
   2. Demostración Técnica
   3. Preguntas y Respuestas
7. Conclusiones y Aprendizajes
8. **Introducción**

**Achi Valentinas** es una quinta restaurante ubicada en la zona de **San Jerónimo, Cusco**, específicamente en la dirección **F463+C4H, Circunvalación Nte., Cusco 08006**. Este establecimiento se ha consolidado como un espacio que ofrece un servicio cálido, con una propuesta gastronómica atractiva, orientada a satisfacer tanto a turistas como a la población local. Con el propósito de optimizar la gestión operativa y administrativa del restaurante, este proyecto tiene como objetivo el diseño e implementación de una base de datos robusta, normalizada y funcional, que permita registrar, controlar y consultar información relevante como reservas, menú, clientes, personal, inventario y proveedores.

A través de este informe, se documentan todas las fases del proyecto: desde el diseño conceptual y lógico de la base de datos, hasta su implementación en un sistema gestor de base de datos (SQL Server), incluyendo la creación de procedimientos almacenados (CRUD) para una gestión dinámica y eficiente de los datos.

1. **Diseño de la Base de Datos**
   1. **Diagrama Entidad-Relación Lógico**
   2. **Diagrama Entidad-Relación Físico**

****

* 1. **Diccionario de Datos**

## Tabla: TUsuario

| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdUsuario | INT | - | PK | Identificador único del usuario |
| NombreUsuario | VARCHAR | 50 | - | Nombre de acceso del usuario |
| Clave | VARCHAR | 100 | - | Contraseña del usuario |
| TipoUsuario | VARCHAR | 50 | - | Rol del usuario (Administrador, Chef) |

## Tabla: TCliente

| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdCliente | INT | - | PK | Identificador único del cliente |
| Nombre | VARCHAR | 100 | - | Nombre del cliente |
| Apellido | VARCHAR | 100 | - | Apellido del cliente |
| Telefono | VARCHAR | 15 | - | Número de contacto |
| Correo | VARCHAR | 100 | - | Correo electrónico |
| Direccion | VARCHAR | 200 | - | Dirección física |

## Tabla: TEmpleado

| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdEmpleado | INT | - | PK | Identificador único del empleado |
| Nombre | VARCHAR | 100 | - | Nombre del empleado |
| Apellido | VARCHAR | 100 | - | Apellido del empleado |
| Cargo | VARCHAR | 50 | - | Puesto que desempeña |
| Telefono | VARCHAR | 15 | - | Número de contacto |
| Correo | VARCHAR | 100 | - | Correo institucional |
| IdUsuario | INT | - | FK | Relación con el usuario asignado |

## Tabla: TPlato

| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdPlato | INT | - | PK | Identificador único del plato |
| Nombre | VARCHAR | 100 | - | Nombre del plato |
| Descripcion | VARCHAR | 255 | - | Detalles o ingredientes |
| Precio | DECIMAL | 10,2 | - | Precio unitario del plato |
| Categoria | VARCHAR | 100 | - | Clasificación del plato |

## Tabla: TMesa

| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdMesa | INT | - | PK | Identificador único de la mesa |
| NumeroMesa | INT | - | - | Número asignado a la mesa |
| Capacidad | INT | - | - | Cantidad de personas que soporta |

## Tabla: TReserva

| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdReserva | INT | - | PK | Identificador único de la reserva |
| Fecha | DATE | - | - | Fecha programada para la reserva |
| Hora | TIME | - | - | Hora de la reserva |
| IdCliente | INT | - | FK | Cliente que realiza la reserva |
| IdMesa | INT | - | FK | Mesa asignada a la reserva |

## Tabla: TPedido

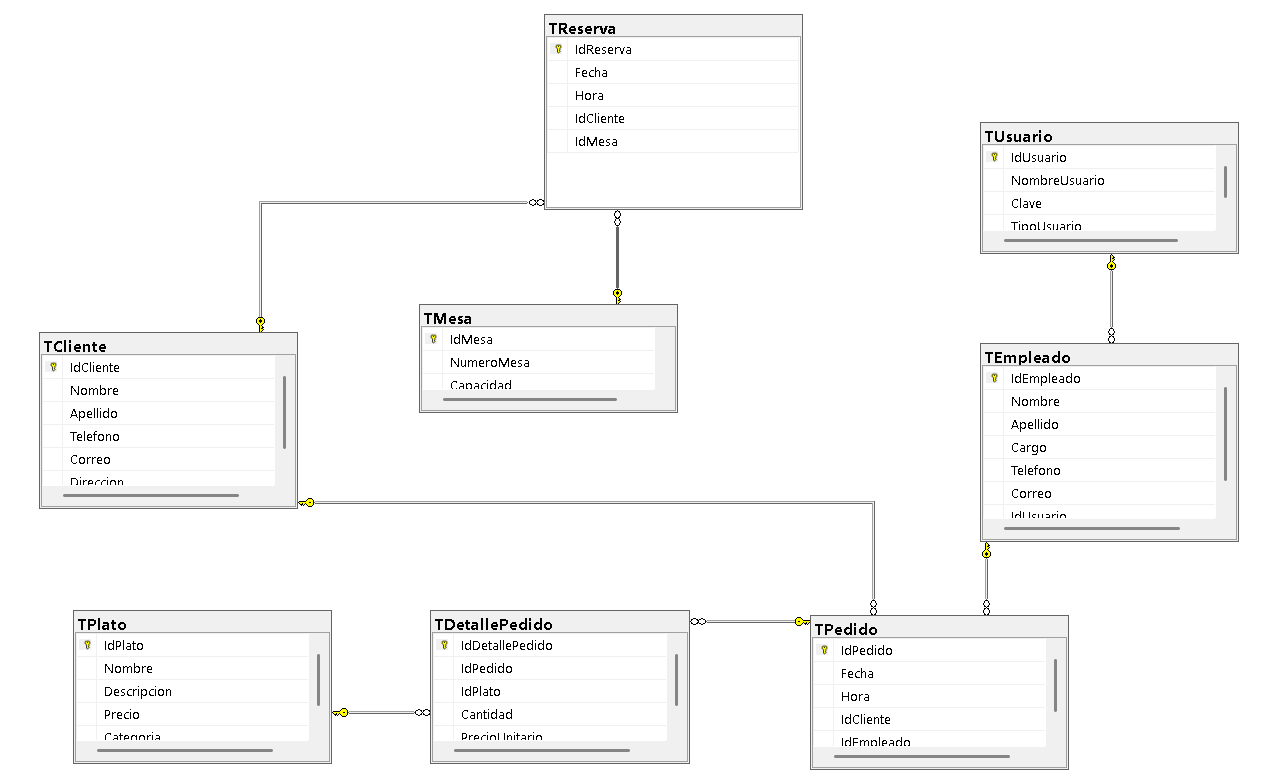
| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdPedido | INT | - | PK | Identificador único del pedido |
| Fecha | DATE | - | - | Fecha del pedido |
| Hora | TIME | - | - | Hora del pedido |
| IdCliente | INT | - | FK | Cliente que hace el pedido |
| IdEmpleado | INT | - | FK | Empleado que atiende el pedido |

## Tabla: TDetallePedido

| Campo | Tipo de Dato | Tamaño | Clave | Descripción |
| --- | --- | --- | --- | --- |
| IdDetallePedido | INT | - | PK | Identificador único del detalle |
| IdPedido | INT | - | FK | Pedido al que pertenece |
| IdPlato | INT | - | FK | Plato solicitado |
| Cantidad | INT | - | - | Número de unidades del plato |
| PrecioUnitario | DECIMAL | 10,2 | - | Precio del plato en el pedido |

* 1. **Justificación del Diseño**

El objetivo principal fue crear una base de datos eficiente, coherente y normalizada que permitiera gestionar las operaciones internas del restaurante "Quinta Valentinas", como reservas, pedidos, empleados y platos ofrecidos. El diseño se basa en un enfoque relacional, priorizando integridad de datos, escalabilidad y facilidad de mantenimiento.

1. **Implementación de la Base de Datos**
   1. **Esquema General de la Base de Datos**
   2. **Creación de Tablas y Restricciones**

**-- Crear la base de datos**

**CREATE DATABASE PROYECTO;**

**GO**

**USE PROYECTO;**

**GO**

**-- Tabla: Usuarios (Incluye tipo directamente)**

**CREATE TABLE TUsuario (**

**IdUsuario INT PRIMARY KEY IDENTITY(1,1),**

**NombreUsuario VARCHAR(50) NOT NULL,**

**Clave VARCHAR(100) NOT NULL,**

**TipoUsuario VARCHAR(50) NOT NULL -- ejemplo: 'Administrador', 'Chef'**

**);**

**-- Tabla: Clientes**

**CREATE TABLE TCliente (**

**IdCliente INT PRIMARY KEY IDENTITY(1,1),**

**Nombre VARCHAR(100) NOT NULL,**

**Apellido VARCHAR(100) NOT NULL,**

**Telefono VARCHAR(15),**

**Correo VARCHAR(100),**

**Direccion VARCHAR(200)**

**);**

**-- Tabla: Empleados**

**CREATE TABLE TEmpleado (**

**IdEmpleado INT PRIMARY KEY IDENTITY(1,1),**

**Nombre VARCHAR(100) NOT NULL,**

**Apellido VARCHAR(100) NOT NULL,**

**Cargo VARCHAR(50),**

**Telefono VARCHAR(15),**

**Correo VARCHAR(100),**

**IdUsuario INT,**

**FOREIGN KEY (IdUsuario) REFERENCES TUsuario(IdUsuario)**

**);**

**-- Tabla: Platos (Incluye categoría directamente)**

**CREATE TABLE TPlato (**

**IdPlato INT PRIMARY KEY IDENTITY(1,1),**

**Nombre VARCHAR(100) NOT NULL,**

**Descripcion VARCHAR(255),**

**Precio DECIMAL(10,2) NOT NULL,**

**Categoria VARCHAR(100) NOT NULL -- ejemplo: 'Entradas', 'Plato de Fondo'**

**);**

**-- Tabla: Mesas**

**CREATE TABLE TMesa (**

**IdMesa INT PRIMARY KEY IDENTITY(1,1),**

**NumeroMesa INT NOT NULL,**

**Capacidad INT NOT NULL**

**);**

**-- Tabla: Reservas**

**CREATE TABLE TReserva (**

**IdReserva INT PRIMARY KEY IDENTITY(1,1),**

**Fecha DATE NOT NULL,**

**Hora TIME NOT NULL,**

**IdCliente INT NOT NULL,**

**IdMesa INT NOT NULL,**

**FOREIGN KEY (IdCliente) REFERENCES TCliente(IdCliente),**

**FOREIGN KEY (IdMesa) REFERENCES TMesa(IdMesa)**

**);**

**-- Tabla: Pedidos**

**CREATE TABLE TPedido (**

**IdPedido INT PRIMARY KEY IDENTITY(1,1),**

**Fecha DATE NOT NULL,**

**Hora TIME NOT NULL,**

**IdCliente INT NOT NULL,**

**IdEmpleado INT NOT NULL,**

**FOREIGN KEY (IdCliente) REFERENCES TCliente(IdCliente),**

**FOREIGN KEY (IdEmpleado) REFERENCES TEmpleado(IdEmpleado)**

**);**

**-- Tabla: Detalles del Pedido**

**CREATE TABLE TDetallePedido (**

**IdDetallePedido INT PRIMARY KEY IDENTITY(1,1),**

**IdPedido INT NOT NULL,**

**IdPlato INT NOT NULL,**

**Cantidad INT NOT NULL,**

**PrecioUnitario DECIMAL(10,2) NOT NULL,**

**FOREIGN KEY (IdPedido) REFERENCES TPedido(IdPedido),**

**FOREIGN KEY (IdPlato) REFERENCES TPlato(IdPlato)**

**);**

* 1. **Procedimientos Almacenados CRUD**

**CRUD para TEmpleados**

**-- LISTAR EMPLEADOS**

**IF OBJECT\_ID('spListarEmpleado') IS NOT NULL**

**DROP PROC spListarEmpleado**

**GO**

**CREATE PROC spListarEmpleado**

**AS**

**BEGIN**

**SELECT IdEmpleado, Nombre, Apellido, Cargo, Telefono, Correo**

**FROM TEmpleado**

**END**

**GO**

**EXEC spListarEmpleado**

**GO**

**-- AGREGAR EMPLEADO**

**IF OBJECT\_ID('spAgregarEmpleado', 'P') IS NOT NULL**

**DROP PROCEDURE spAgregarEmpleado**

**GO**

**CREATE PROCEDURE spAgregarEmpleado**

**@Nombre VARCHAR(100),**

**@Apellido VARCHAR(100),**

**@Cargo VARCHAR(50),**

**@Telefono VARCHAR(15),**

**@Correo VARCHAR(100),**

**@IdUsuario INT**

**AS**

**BEGIN**

**-- Verificar si el Usuario asociado al empleado existe**

**IF EXISTS (SELECT IdUsuario FROM TUsuario WHERE IdUsuario = @IdUsuario)**

**BEGIN**

**INSERT INTO TEmpleado (Nombre, Apellido, Cargo, Telefono, Correo, IdUsuario)**

**VALUES (@Nombre, @Apellido, @Cargo, @Telefono, @Correo, @IdUsuario)**

**SELECT CodError = 0, Mensaje = 'Correcto, empleado insertado'**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: El Usuario no existe'**

**END**

**GO**

**-- ELIMINAR EMPLEADO**

**IF OBJECT\_ID('spEliminarEmpleado', 'P') IS NOT NULL**

**DROP PROC spEliminarEmpleado**

**GO**

**CREATE PROC spEliminarEmpleado**

**@IdEmpleado INT**

**AS**

**BEGIN**

**-- Verificar si el empleado existe**

**IF EXISTS (SELECT IdEmpleado FROM TEmpleado WHERE IdEmpleado = @IdEmpleado)**

**BEGIN**

**-- Verificar si el empleado está relacionado con algún pedido**

**IF EXISTS (SELECT IdEmpleado FROM TPedido WHERE IdEmpleado = @IdEmpleado)**

**BEGIN**

**-- Si está relacionado, no eliminar y dar mensaje de error**

**SELECT CodError = 1, Mensaje = 'Error: El empleado está relacionado con pedidos y no se puede eliminar'**

**END**

**ELSE**

**BEGIN**

**-- Eliminar el empleado si no está relacionado**

**DELETE FROM TEmpleado WHERE IdEmpleado = @IdEmpleado**

**SELECT CodError = 0, Mensaje = 'Correcto, empleado eliminado'**

**END**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: El empleado no existe'**

**END**

**GO**

**-- ACTUALIZAR EMPLEADO**

**IF OBJECT\_ID('spActualizarEmpleado', 'P') IS NOT NULL**

**DROP PROC spActualizarEmpleado**

**GO**

**CREATE PROC spActualizarEmpleado**

**@IdEmpleado INT,**

**@NuevoNombre VARCHAR(100),**

**@NuevoApellido VARCHAR(100),**

**@NuevoCargo VARCHAR(50),**

**@NuevoTelefono VARCHAR(15),**

**@NuevoCorreo VARCHAR(100),**

**@NuevoIdUsuario INT**

**AS**

**BEGIN**

**-- Verificar si el empleado existe**

**IF EXISTS (SELECT IdEmpleado FROM TEmpleado WHERE IdEmpleado = @IdEmpleado)**

**BEGIN**

**-- Verificar si el nuevo Usuario asociado al empleado existe**

**IF EXISTS (SELECT IdUsuario FROM TUsuario WHERE IdUsuario = @NuevoIdUsuario)**

**BEGIN**

**-- Actualizar los campos del empleado**

**UPDATE TEmpleado**

**SET Nombre = @NuevoNombre,**

**Apellido = @NuevoApellido,**

**Cargo = @NuevoCargo,**

**Telefono = @NuevoTelefono,**

**Correo = @NuevoCorreo,**

**IdUsuario = @NuevoIdUsuario**

**WHERE IdEmpleado = @IdEmpleado**

**SELECT CodError = 0, Mensaje = 'Empleado actualizado correctamente'**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: El nuevo Usuario no existe'**

**END**

**ELSE**

**BEGIN**

**SELECT CodError = 1, Mensaje = 'Error: El empleado no existe'**

**END**

**END**

**GO**

**-- BUSCAR EMPLEADO**

**IF OBJECT\_ID('spBuscarEmpleado', 'P') IS NOT NULL**

**DROP PROC spBuscarEmpleado**

**GO**

**CREATE PROC spBuscarEmpleado**

**@Texto VARCHAR(50),**

**@Criterio VARCHAR(10)**

**AS**

**BEGIN**

**IF (@Criterio = 'IdEmpleado')**

**-- Búsqueda exacta por IdEmpleado**

**SELECT IdEmpleado, Nombre, Apellido, Cargo, Telefono, Correo**

**FROM TEmpleado**

**WHERE IdEmpleado = @Texto**

**ELSE IF (@Criterio = 'Nombre')**

**-- Búsqueda sensitiva por Nombre**

**SELECT IdEmpleado, Nombre, Apellido, Cargo, Telefono, Correo**

**FROM TEmpleado**

**WHERE Nombre LIKE '%' + @Texto + '%'**

**ELSE IF (@Criterio = 'Apellido')**

**-- Búsqueda sensitiva por Apellido**

**SELECT IdEmpleado, Nombre, Apellido, Cargo, Telefono, Correo**

**FROM TEmpleado**

**WHERE Apellido LIKE '%' + @Texto + '%'**

**ELSE IF (@Criterio = 'Cargo')**

**-- Búsqueda sensitiva por Cargo**

**SELECT IdEmpleado, Nombre, Apellido, Cargo, Telefono, Correo**

**FROM TEmpleado**

**WHERE Cargo LIKE '%' + @Texto + '%'**

**END**

**GO**

**CRUD PARA TPlatos**

**-- LISTAR PLATOS**

**IF OBJECT\_ID('spListarPlato') IS NOT NULL**

**DROP PROC spListarPlato**

**GO**

**CREATE PROC spListarPlato**

**AS**

**BEGIN**

**SELECT IdPlato, Nombre, Descripcion, Precio, Categoria**

**FROM TPlato**

**END**

**GO**

**EXEC spListarPlato**

**GO**

**-- AGREGAR PLATO**

**IF OBJECT\_ID('spAgregarPlato', 'P') IS NOT NULL**

**DROP PROCEDURE spAgregarPlato**

**GO**

**CREATE PROCEDURE spAgregarPlato**

**@Nombre VARCHAR(100),**

**@Descripcion VARCHAR(255),**

**@Precio DECIMAL(10,2),**

**@Categoria VARCHAR(100)**

**AS**

**BEGIN**

**-- Verificar si el Nombre del Plato no está duplicado**

**IF NOT EXISTS (SELECT Nombre FROM TPlato WHERE Nombre = @Nombre)**

**BEGIN**

**INSERT INTO TPlato (Nombre, Descripcion, Precio, Categoria)**

**VALUES (@Nombre, @Descripcion, @Precio, @Categoria)**

**SELECT CodError = 0, Mensaje = 'Correcto, plato insertado'**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: Plato duplicado'**

**END**

**GO**

**-- ELIMINAR PLATO**

**IF OBJECT\_ID('spEliminarPlato', 'P') IS NOT NULL**

**DROP PROC spEliminarPlato**

**GO**

**CREATE PROC spEliminarPlato**

**@IdPlato INT**

**AS**

**BEGIN**

**-- Verificar si el plato existe**

**IF EXISTS (SELECT IdPlato FROM TPlato WHERE IdPlato = @IdPlato)**

**BEGIN**

**-- Verificar si el plato está relacionado con algún detalle de pedido**

**IF EXISTS (SELECT IdPlato FROM TDetallePedido WHERE IdPlato = @IdPlato)**

**BEGIN**

**-- Eliminar primero los detalles del pedido que contienen el plato**

**DELETE FROM TDetallePedido WHERE IdPlato = @IdPlato**

**SELECT CodError = 0, Mensaje = 'Detalles de pedido eliminados'**

**END**

**-- Ahora eliminar el plato**

**DELETE FROM TPlato WHERE IdPlato = @IdPlato**

**SELECT CodError = 0, Mensaje = 'Correcto, plato eliminado'**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: El plato no existe'**

**END**

**GO**

**-- ACTUALIZAR PLATO**

**IF OBJECT\_ID('spActualizarPlato', 'P') IS NOT NULL**

**DROP PROC spActualizarPlato**

**GO**

**CREATE PROC spActualizarPlato**

**@IdPlato INT,**

**@NuevoNombre VARCHAR(100),**

**@NuevaDescripcion VARCHAR(255),**

**@NuevoPrecio DECIMAL(10,2),**

**@NuevaCategoria VARCHAR(100)**

**AS**

**BEGIN**

**-- Verificar si el plato existe**

**IF EXISTS (SELECT IdPlato FROM TPlato WHERE IdPlato = @IdPlato)**

**BEGIN**

**-- Actualizar los campos del plato**

**UPDATE TPlato**

**SET Nombre = @NuevoNombre,**

**Descripcion = @NuevaDescripcion,**

**Precio = @NuevoPrecio,**

**Categoria = @NuevaCategoria**

**WHERE IdPlato = @IdPlato**

**SELECT CodError = 0, Mensaje = 'Plato actualizado correctamente'**

**END**

**ELSE**

**BEGIN**

**SELECT CodError = 1, Mensaje = 'Error: El plato no existe'**

**END**

**END**

**GO**

**-- BUSCAR PLATO**

**IF OBJECT\_ID('spBuscarPlato', 'P') IS NOT NULL**

**DROP PROC spBuscarPlato**

**GO**

**CREATE PROC spBuscarPlato**

**@Texto VARCHAR(50),**

**@Criterio VARCHAR(10)**

**AS**

**BEGIN**

**IF (@Criterio = 'IdPlato')**

**-- Búsqueda exacta por IdPlato**

**SELECT IdPlato, Nombre, Descripcion, Precio, Categoria**

**FROM TPlato**

**WHERE IdPlato = @Texto**

**ELSE IF (@Criterio = 'Nombre')**

**-- Búsqueda sensitiva por Nombre**

**SELECT IdPlato, Nombre, Descripcion, Precio, Categoria**

**FROM TPlato**

**WHERE Nombre LIKE '%' + @Texto + '%'**

**ELSE IF (@Criterio = 'Categoria')**

**-- Búsqueda sensitiva por Categoria**

**SELECT IdPlato, Nombre, Descripcion, Precio**

**FROM TPlato**

**WHERE Categoria LIKE '%' + @Texto + '%'**

**END**

**GO**

**--Comprobacion del codigo**

**-- Ejecutar el procedimiento para listar todos los platos**

**EXEC spListarPlato**

**GO**

**-- Ejecutar el procedimiento para agregar un plato**

**EXEC spAgregarPlato**

**@Nombre = 'Pizza Margherita',**

**@Descripcion = 'Pizza con tomate, mozzarella y albahaca',**

**@Precio = 15.99,**

**@Categoria = 'Plato de Fondo'**

**GO**

**-- Ejecutar el procedimiento para eliminar un plato con IdPlato = 1**

**EXEC spEliminarPlato @IdPlato = 1**

**GO**

**-- Ejecutar el procedimiento para actualizar un plato con IdPlato = 1**

**EXEC spActualizarPlato**

**@IdPlato = 1,**

**@NuevoNombre = 'Pizza Margherita Deluxe',**

**@NuevaDescripcion = 'Pizza con tomate, mozzarella, albahaca y aceite de oliva extra virgen',**

**@NuevoPrecio = 18.99,**

**@NuevaCategoria = 'Plato de Fondo'**

**GO**

**-- Ejecutar el procedimiento para buscar un plato por Nombre**

**EXEC spBuscarPlato**

**@Texto = 'Pizza',**

**@Criterio = 'Nombre'**

**GO**

**-- Ejecutar el procedimiento para buscar un plato por Categoria**

**EXEC spBuscarPlato**

**@Texto = 'Plato de Fondo',**

**@Criterio = 'Categoria'**

**GO**

**CRUD para TUsuarios**

**-- LISTAR USUARIOS**

**IF OBJECT\_ID('spListarUsuario') IS NOT NULL**

**DROP PROC spListarUsuario**

**GO**

**CREATE PROC spListarUsuario**

**AS**

**BEGIN**

**SELECT IdUsuario, NombreUsuario, TipoUsuario**

**FROM TUsuario**

**END**

**GO**

**EXEC spListarUsuario**

**GO**

**-- AGREGAR USUARIO**

**IF OBJECT\_ID('spAgregarUsuario', 'P') IS NOT NULL**

**DROP PROCEDURE spAgregarUsuario**

**GO**

**CREATE PROCEDURE spAgregarUsuario**

**@NombreUsuario VARCHAR(50),**

**@Clave VARCHAR(100),**

**@TipoUsuario VARCHAR(50)**

**AS**

**BEGIN**

**-- Verificar si el NombreUsuario no está duplicado**

**IF NOT EXISTS (SELECT NombreUsuario FROM TUsuario WHERE NombreUsuario = @NombreUsuario)**

**BEGIN**

**INSERT INTO TUsuario (NombreUsuario, Clave, TipoUsuario)**

**VALUES (@NombreUsuario, @Clave, @TipoUsuario)**

**SELECT CodError = 0, Mensaje = 'Correcto, usuario insertado'**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: NombreUsuario duplicado'**

**END**

**GO**

**-- ELIMINAR USUARIO**

**IF OBJECT\_ID('spEliminarUsuario', 'P') IS NOT NULL**

**DROP PROC spEliminarUsuario**

**GO**

**CREATE PROC spEliminarUsuario**

**@IdUsuario INT**

**AS**

**BEGIN**

**-- Verificar si el usuario existe**

**IF EXISTS (SELECT IdUsuario FROM TUsuario WHERE IdUsuario = @IdUsuario)**

**BEGIN**

**-- Verificar si el usuario está relacionado con algún empleado**

**IF EXISTS (SELECT IdUsuario FROM TEmpleado WHERE IdUsuario = @IdUsuario)**

**BEGIN**

**-- Si está relacionado, no eliminar y dar mensaje de error**

**SELECT CodError = 1, Mensaje = 'Error: El usuario está relacionado con empleados y no se puede eliminar'**

**END**

**ELSE**

**BEGIN**

**-- Eliminar el usuario si no está relacionado**

**DELETE FROM TUsuario WHERE IdUsuario = @IdUsuario**

**SELECT CodError = 0, Mensaje = 'Correcto, usuario eliminado'**

**END**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: El usuario no existe'**

**END**

**GO**

**-- ACTUALIZAR USUARIO**

**IF OBJECT\_ID('spActualizarUsuario', 'P') IS NOT NULL**

**DROP PROC spActualizarUsuario**

**GO**

**CREATE PROC spActualizarUsuario**

**@IdUsuario INT,**

**@NuevoNombreUsuario VARCHAR(50),**

**@NuevaClave VARCHAR(100),**

**@NuevoTipoUsuario VARCHAR(50)**

**AS**

**BEGIN**

**-- Verificar si el usuario existe**

**IF EXISTS (SELECT IdUsuario FROM TUsuario WHERE IdUsuario = @IdUsuario)**

**BEGIN**

**-- Actualizar los campos del usuario**

**UPDATE TUsuario**

**SET NombreUsuario = @NuevoNombreUsuario,**

**Clave = @NuevaClave,**

**TipoUsuario = @NuevoTipoUsuario**

**WHERE IdUsuario = @IdUsuario**

**SELECT CodError = 0, Mensaje = 'Usuario actualizado correctamente'**

**END**

**ELSE**

**BEGIN**

**SELECT CodError = 1, Mensaje = 'Error: El usuario no existe'**

**END**

**END**

**GO**

**-- BUSCAR USUARIO**

**IF OBJECT\_ID('spBuscarUsuario', 'P') IS NOT NULL**

**DROP PROC spBuscarUsuario**

**GO**

**CREATE PROC spBuscarUsuario**

**@Texto VARCHAR(50),**

**@Criterio VARCHAR(10)**

**AS**

**BEGIN**

**IF (@Criterio = 'IdUsuario')**

**-- Búsqueda exacta por IdUsuario**

**SELECT IdUsuario, NombreUsuario, TipoUsuario**

**FROM TUsuario**

**WHERE IdUsuario = @Texto**

**ELSE IF (@Criterio = 'NombreUsuario')**

**-- Búsqueda sensitiva por NombreUsuario**

**SELECT IdUsuario, NombreUsuario, TipoUsuario**

**FROM TUsuario**

**WHERE NombreUsuario LIKE '%' + @Texto + '%'**

**ELSE IF (@Criterio = 'TipoUsuario')**

**-- Búsqueda sensitiva por TipoUsuario**

**SELECT IdUsuario, NombreUsuario**

**FROM TUsuario**

**WHERE TipoUsuario LIKE '%' + @Texto + '%'**

**END**

**GO**

**CRUD PARA TClientes**

**-- LISTAR CLIENTE**

**IF OBJECT\_ID('spListarCliente') IS NOT NULL**

**DROP PROC spListarCliente**

**GO**

**CREATE PROC spListarCliente**

**AS**

**BEGIN**

**SELECT IdCliente, Nombre, Apellido, Direccion**

**FROM TCliente**

**END**

**GO**

**EXEC spListarCliente**

**GO**

**-- AGREGAR CLIENTE**

**IF OBJECT\_ID('spAgregarCliente', 'P') IS NOT NULL**

**DROP PROCEDURE spAgregarCliente**

**GO**

**CREATE PROCEDURE spAgregarCliente**

**@IdCliente INT,**

**@Nombre VARCHAR(100),**

**@Apellido VARCHAR(100),**

**@Telefono VARCHAR(15),**

**@Correo VARCHAR(100),**

**@Direccion VARCHAR(200)**

**AS**

**BEGIN**

**-- Verificar si el IdCliente no está duplicado**

**IF NOT EXISTS (SELECT IdCliente FROM TCliente WHERE IdCliente = @IdCliente)**

**BEGIN**

**INSERT INTO TCliente (Nombre, Apellido, Telefono, Correo, Direccion)**

**VALUES (@Nombre, @Apellido, @Telefono, @Correo, @Direccion)**

**SELECT CodError = 0, Mensaje = 'Correcto, cliente insertado'**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: IdCliente duplicado'**

**END**

**GO**

**-- ELIMINAR CLIENTE**

**IF OBJECT\_ID('spEliminarCliente', 'P') IS NOT NULL**

**DROP PROC spEliminarCliente**

**GO**

**CREATE PROC spEliminarCliente**

**@IdCliente INT**

**AS**

**BEGIN**

**-- Verificar si el cliente existe**

**IF EXISTS (SELECT IdCliente FROM TCliente WHERE IdCliente = @IdCliente)**

**BEGIN**

**-- Verificar si el cliente está relacionado con alguna reserva**

**IF EXISTS (SELECT IdCliente FROM TReserva WHERE IdCliente = @IdCliente)**

**BEGIN**

**-- Si está relacionado, no eliminar y dar mensaje de error**

**SELECT CodError = 1, Mensaje = 'Error: El cliente está relacionado con reservas y no se puede eliminar'**

**END**

**ELSE**

**BEGIN**

**-- Eliminar el cliente si no está relacionado**

**DELETE FROM TCliente WHERE IdCliente = @IdCliente**

**SELECT CodError = 0, Mensaje = 'Correcto, cliente eliminado'**

**END**

**END**

**ELSE**

**SELECT CodError = 1, Mensaje = 'Error: El cliente no existe'**

**END**

**GO**

**-- ACTUALIZAR CLIENTE**

**IF OBJECT\_ID('spActualizarCliente', 'P') IS NOT NULL**

**DROP PROC spActualizarCliente**

**GO**

**CREATE PROC spActualizarCliente**

**@IdCliente INT,**

**@NuevoNombre VARCHAR(100),**

**@NuevoApellido VARCHAR(100),**

**@NuevoTelefono VARCHAR(15),**

**@NuevoCorreo VARCHAR(100),**

**@NuevaDireccion VARCHAR(200)**

**AS**

**BEGIN**

**-- Verificar si el cliente existe**

**IF EXISTS (SELECT IdCliente FROM TCliente WHERE IdCliente = @IdCliente)**

**BEGIN**

**-- Actualizar los campos del cliente**

**UPDATE TCliente**

**SET Nombre = @NuevoNombre,**

**Apellido = @NuevoApellido,**

**Telefono = @NuevoTelefono,**

**Correo = @NuevoCorreo,**

**Direccion = @NuevaDireccion**

**WHERE IdCliente = @IdCliente**

**SELECT CodError = 0, Mensaje = 'Cliente actualizado correctamente'**

**END**

**ELSE**

**BEGIN**

**SELECT CodError = 1, Mensaje = 'Error: El cliente no existe'**

**END**

**END**

**GO**

**-- BUSCAR CLIENTE**

**IF OBJECT\_ID('spBuscarCliente', 'P') IS NOT NULL**

**DROP PROC spBuscarCliente**

**GO**

**CREATE PROC spBuscarCliente**

**@Texto VARCHAR(50),**

**@Criterio VARCHAR(10)**

**AS**

**BEGIN**

**IF (@Criterio = 'IdCliente')**

**-- Búsqueda exacta por IdCliente**

**SELECT IdCliente, Nombre, Apellido, Direccion**

**FROM TCliente**

**WHERE IdCliente = @Texto**

**ELSE IF (@Criterio = 'Nombre')**

**-- Búsqueda sensitiva por Nombre**

**SELECT Nombre, Apellido, Direccion**

**FROM TCliente**

**WHERE Nombre LIKE '%' + @Texto + '%'**

**ELSE IF (@Criterio = 'Apellido')**

**-- Búsqueda sensitiva por Apellido**

**SELECT Nombre, Apellido, Direccion**

**FROM TCliente**

**WHERE Apellido LIKE '%' + @Texto + '%'**

**ELSE IF (@Criterio = 'Direccion')**

**-- Búsqueda sensitiva por Dirección**

**SELECT Nombre, Apellido, Direccion**

**FROM TCliente**

**WHERE Direccion LIKE '%' + @Texto + '%'**

**END**

**GO**

**3.4 DATOS DE PRUEBA**

**-- Inserciones en la tabla TUsuario**

**INSERT INTO TUsuario (NombreUsuario, Clave, TipoUsuario) VALUES**

**('admin1', 'adminpass', 'Administrador'),**

**('chef1', 'chefpass', 'Chef'),**

**('mesero1', 'meseropass', 'Mesero'),**

**('admin2', 'adminpass2', 'Administrador'),**

**('chef2', 'chefpass2', 'Chef'),**

**('mesero2', 'meseropass2', 'Mesero'),**

**('admin3', 'adminpass3', 'Administrador'),**

**('chef3', 'chefpass3', 'Chef'),**

**('mesero3', 'meseropass3', 'Mesero'),**

**('admin4', 'adminpass4', 'Administrador'),**

**('chef4', 'chefpass4', 'Chef'),**

**('mesero4', 'meseropass4', 'Mesero'),**

**('admin5', 'adminpass5', 'Administrador'),**

**('chef5', 'chefpass5', 'Chef'),**

**('mesero5', 'meseropass5', 'Mesero'),**

**('admin6', 'adminpass6', 'Administrador'),**

**('chef6', 'chefpass6', 'Chef'),**

**('mesero6', 'meseropass6', 'Mesero'),**

**('admin7', 'adminpass7', 'Administrador'),**

**('chef7', 'chefpass7', 'Chef');**

**-- Inserciones en la tabla TCliente**

**INSERT INTO TCliente (Nombre, Apellido, Telefono, Correo, Direccion) VALUES**

**('Carlos', 'Ramirez', '987654321', 'carlos@email.com', 'Av. Central 123'),**

**('Lucía', 'Pérez', '912345678', 'lucia@email.com', 'Calle Lima 456'),**

**('Juan', 'Gonzales', '987654322', 'juan@email.com', 'Av. Morales 789'),**

**('María', 'Lopez', '912345679', 'maria@email.com', 'Calle Balta 101'),**

**('Pedro', 'Martínez', '987654323', 'pedro@email.com', 'Av. San Martín 202'),**

**('Ana', 'Rodríguez', '912345680', 'ana@email.com', 'Calle Jirón 303'),**

**('Luis', 'Fernández', '987654324', 'luis@email.com', 'Av. Grau 404'),**

**('Sofía', 'García', '912345681', 'sofia@email.com', 'Calle Ayacucho 505'),**

**('Miguel', 'Díaz', '987654325', 'miguel@email.com', 'Av. Arequipa 606'),**

**('Laura', 'González', '912345682', 'laura@email.com', 'Calle Santa Teresa 707'),**

**('Carlos', 'Ramírez', '987654326', 'carlosr@email.com', 'Calle Puno 808'),**

**('Gloria', 'Sánchez', '912345683', 'gloria@email.com', 'Av. Javier Prado 909'),**

**('José', 'Torres', '987654327', 'jose@email.com', 'Calle Cajamarca 1010'),**

**('Elena', 'Martínez', '912345684', 'elena@email.com', 'Av. Belén 1111'),**

**('David', 'López', '987654328', 'david@email.com', 'Calle San Pedro 1212'),**

**('Gabriela', 'Morales', '912345685', 'gabriela@email.com', 'Av. Colón 1313'),**

**('Andrés', 'Hernández', '987654329', 'andres@email.com', 'Calle San José 1414'),**

**('Rosa', 'Ramírez', '912345686', 'rosa@email.com', 'Av. Santa Cruz 1515'),**

**('Carlos', 'Cruz', '987654330', 'carloscruz@email.com', 'Calle Los Álamos 1616'),**

**('Sandra', 'Fernández', '912345687', 'sandra@email.com', 'Av. El Sol 1717');**

**-- Inserciones en la tabla TEmpleado**

**INSERT INTO TEmpleado (Nombre, Apellido, Cargo, Telefono, Correo, IdUsuario) VALUES**

**('Juan', 'Gonzales', 'Administrador', '999111222', 'juan@quinta.com', 1),**

**('María', 'Lopez', 'Mesero', '999111223', 'maria@quinta.com', 2),**

**('José', 'García', 'Chef', '999111224', 'jose@quinta.com', 3),**

**('Luis', 'Martínez', 'Administrador', '999111225', 'luis@quinta.com', 4),**

**('Ana', 'Rodríguez', 'Mesero', '999111226', 'ana@quinta.com', 5),**

**('Miguel', 'Fernández', 'Chef', '999111227', 'miguel@quinta.com', 6),**

**('Carlos', 'Díaz', 'Administrador', '999111228', 'carlos@quinta.com', 7),**

**('Laura', 'González', 'Mesero', '999111229', 'laura@quinta.com', 8),**

**('Elena', 'Sánchez', 'Chef', '999111230', 'elena@quinta.com', 9),**

**('David', 'Torres', 'Administrador', '999111231', 'david@quinta.com', 10),**

**('Gabriela', 'López', 'Mesero', '999111232', 'gabriela@quinta.com', 11),**

**('Andrés', 'Martínez', 'Chef', '999111233', 'andres@quinta.com', 12),**

**('Rosa', 'Ramírez', 'Administrador', '999111234', 'rosa@quinta.com', 13),**

**('Carlos', 'Hernández', 'Mesero', '999111235', 'carlos@quinta.com', 14),**

**('Sandra', 'Cruz', 'Chef', '999111236', 'sandra@quinta.com', 15),**

**('José', 'Sánchez', 'Administrador', '999111237', 'jose@quinta.com', 16),**

**('Marta', 'García', 'Mesero', '999111238', 'marta@quinta.com', 17),**

**('Juan', 'Martínez', 'Chef', '999111239', 'juan@quinta.com', 18),**

**('Gloria', 'López', 'Administrador', '999111240', 'gloria@quinta.com', 19),**

**('Sofía', 'Fernández', 'Mesero', '999111241', 'sofia@quinta.com', 20);**

**SELECT \* FROM TEmpleado;**

**-- Inserciones en la tabla TPlato**

**INSERT INTO TPlato (Nombre, Descripcion, Precio, Categoria) VALUES**

**('Sopa de Pollo', 'Sopa de pollo con verduras frescas', 15.50, 'Entradas'),**

**('Ensalada César', 'Ensalada con lechuga, pollo y aderezo César', 10.00, 'Entradas'),**

**('Lomo Saltado', 'Lomo de res con papas fritas y arroz', 20.00, 'Plato de Fondo'),**

**('Arroz Chaufa', 'Arroz frito con pollo y vegetales', 18.75, 'Plato de Fondo'),**

**('Ceviche', 'Pescado marinado con limón, cebolla y cilantro', 12.50, 'Entradas'),**

**('Pasta Alfredo', 'Pasta con salsa cremosa de queso', 10.00, 'Plato de Fondo'),**

**('Pescado a la Parrilla', 'Pescado fresco a la parrilla con guarnición', 20.00, 'Plato de Fondo'),**

**('Tacos de Pollo', 'Tacos con pollo, lechuga y salsa', 18.75, 'Entradas'),**

**('Pizza Margarita', 'Pizza con salsa de tomate, queso y albahaca', 12.50, 'Plato de Fondo'),**

**('Sándwich Club', 'Sándwich con jamón, pollo, lechuga y tomate', 15.50, 'Entradas'),**

**('Sopa Criolla', 'Sopa con carne de res, papas y fideos', 10.00, 'Entradas'),**

**('Chicharrón de Pollo', 'Pollo empanado con papas fritas', 20.00, 'Plato de Fondo'),**

**('Lasaña', 'Lasaña con carne molida y salsa bechamel', 18.75, 'Plato de Fondo'),**

**('Tiradito de Pescado', 'Pescado crudo con salsa picante', 12.50, 'Entradas'),**

**('Pollo a la Brasa', 'Pollo asado con papas y ensalada', 15.50, 'Plato de Fondo'),**

**('Paella', 'Arroz con mariscos y vegetales', 10.00, 'Plato de Fondo'),**

**('Chupe de Camarones', 'Sopa de camarones con papa y leche', 20.00, 'Plato de Fondo'),**

**('Arroz con Mariscos', 'Arroz con mariscos frescos', 18.75, 'Plato de Fondo'),**

**('Sushi', 'Rollos de sushi con pescado y aguacate', 12.50, 'Entradas'),**

**('Lomo de Cerdo', 'Lomo de cerdo al horno con ensalada', 15.50, 'Plato de Fondo');**

**-- Inserciones en la tabla TMesa**

**INSERT INTO TMesa (NumeroMesa, Capacidad) VALUES**

**(1, 4),**

**(2, 2),**

**(3, 6),**

**(4, 4),**

**(5, 8),**

**(6, 2),**

**(7, 4),**

**(8, 6),**

**(9, 2),**

**(10, 4),**

**(11, 6),**

**(12, 8),**

**(13, 4),**

**(14, 2),**

**(15, 4),**

**(16, 6),**

**(17, 2),**

**(18, 4),**

**(19, 6),**

**(20, 4);**

**-- Inserciones en la tabla TReserva**

**INSERT INTO TReserva (Fecha, Hora, IdCliente, IdMesa) VALUES**

**('2025-06-01', '12:30:00', 1, 1),**

**('2025-06-02', '13:00:00', 2, 2),**

**('2025-06-03', '14:15:00', 3, 3),**

**('2025-06-04', '15:45:00', 4, 4),**

**('2025-06-05', '16:00:00', 5, 5),**

**('2025-06-06', '17:20:00', 6, 6),**

**('2025-06-07', '18:05:00', 7, 7),**

**('2025-06-08', '19:30:00', 8, 8),**

**('2025-06-09', '20:10:00', 9, 9),**

**('2025-06-10', '21:25:00', 10, 10),**

**('2025-06-11', '12:45:00', 11, 11),**

**('2025-06-12', '14:00:00', 12, 12),**

**('2025-06-13', '15:30:00', 13, 13),**

**('2025-06-14', '16:15:00', 14, 14),**

**('2025-06-15', '17:50:00', 15, 15),**

**('2025-06-16', '18:25:00', 16, 16),**

**('2025-06-17', '19:00:00', 17, 17),**

**('2025-06-18', '20:30:00', 18, 18),**

**('2025-06-19', '21:10:00', 19, 19),**

**('2025-06-20', '22:00:00', 20, 20);**

**-- Inserciones en la tabla TPedido**

**INSERT INTO TPedido (Fecha, Hora, IdCliente, IdEmpleado) VALUES**

**('2025-06-01', '12:30:00', 1, 1),**

**('2025-06-02', '13:00:00', 2, 2),**

**('2025-06-03', '14:15:00', 3, 3),**

**('2025-06-04', '15:45:00', 4, 4),**

**('2025-06-05', '16:00:00', 5, 5),**

**('2025-06-06', '17:20:00', 6, 6),**

**('2025-06-07', '18:05:00', 7, 7),**

**('2025-06-08', '19:30:00', 8, 8),**

**('2025-06-09', '20:10:00', 9, 9),**

**('2025-06-10', '21:25:00', 10, 10),**

**('2025-06-11', '12:45:00', 11, 11),**

**('2025-06-12', '14:00:00', 12, 12),**

**('2025-06-13', '15:30:00', 13, 13),**

**('2025-06-14', '16:15:00', 14, 14),**

**('2025-06-15', '17:50:00', 15, 15),**

**('2025-06-16', '18:25:00', 16, 16),**

**('2025-06-17', '19:00:00', 17, 17),**

**('2025-06-18', '20:30:00', 18, 18),**

**('2025-06-19', '21:10:00', 19, 19),**

**('2025-06-20', '22:00:00', 20, 20);**

**-- Inserciones en la tabla TDetallePedido**

**INSERT INTO TDetallePedido (IdPedido, IdPlato, Cantidad, PrecioUnitario) VALUES**

**(1, 1, 2, 15.50),**

**(2, 2, 1, 10.00),**

**(3, 3, 1, 20.00),**

**(4, 4, 2, 18.75),**

**(5, 5, 1, 12.50),**

**(6, 2, 3, 10.00),**

**(7, 1, 1, 15.50),**

**(8, 3, 2, 20.00),**

**(9, 4, 1, 18.75),**

**(10, 5, 2, 12.50),**

**(11, 3, 1, 20.00),**

**(12, 4, 2, 18.75),**

**(13, 5, 1, 12.50),**

**(14, 1, 3, 15.50),**

**(15, 2, 1, 10.00),**

**(16, 3, 2, 20.00),**

**(17, 4, 1, 18.75),**

**(18, 5, 2, 12.50),**

**(19, 1, 1, 15.50),**

**(20, 2, 3, 10.00);**

1. **Ejemplos de Uso de Procedimientos Almacenados**

**4.1 Ejemplos para TEmpleados**

**--PRUEBAS DE EMPLEADOS**

**-- Comprobar LISTAR EMPLEADOS**

**EXEC spListarEmpleado**

**GO**

**-- Comprobar AGREGAR EMPLEADO**

**EXEC spAgregarEmpleado**

**@Nombre = 'Juan',**

**@Apellido = 'Perez',**

**@Cargo = 'Chef',**

**@Telefono = '987654321',**

**@Correo = 'juanperez@email.com',**

**@IdUsuario = 1**

**GO**

**-- Comprobar ELIMINAR EMPLEADO**

**EXEC spEliminarEmpleado**

**@IdEmpleado = 1**

**GO**

**-- Comprobar ACTUALIZAR EMPLEADO**

**EXEC spActualizarEmpleado**

**@IdEmpleado = 1,**

**@NuevoNombre = 'Carlos',**

**@NuevoApellido = 'Gonzales',**

**@NuevoCargo = 'Mesero',**

**@NuevoTelefono = '987654322',**

**@NuevoCorreo = 'carlosgonzales@email.com',**

**@NuevoIdUsuario = 2**

**GO**

**-- Comprobar BUSCAR EMPLEADO (por Nombre)**

**EXEC spBuscarEmpleado**

**@Texto = 'Carlos',**

**@Criterio = 'Nombre'**

**GO**

**-- Comprobar BUSCAR EMPLEADO (por Cargo)**

**EXEC spBuscarEmpleado**

**@Texto = 'Chef',**

**@Criterio = 'Cargo'**

**GO**

**4.2 Uso en TPlatos**

**--Comprobacion del codigo**

**-- Ejecutar el procedimiento para listar todos los platos**

**EXEC spListarPlato**

**GO**

**-- Ejecutar el procedimiento para agregar un plato**

**EXEC spAgregarPlato**

**@Nombre = 'Pizza Margherita',**

**@Descripcion = 'Pizza con tomate, mozzarella y albahaca',**

**@Precio = 15.99,**

**@Categoria = 'Plato de Fondo'**

**GO**

**-- Ejecutar el procedimiento para eliminar un plato con IdPlato = 1**

**EXEC spEliminarPlato @IdPlato = 1**

**GO**

**-- Ejecutar el procedimiento para actualizar un plato con IdPlato = 1**

**EXEC spActualizarPlato**

**@IdPlato = 1,**

**@NuevoNombre = 'Pizza Margherita Deluxe',**

**@NuevaDescripcion = 'Pizza con tomate, mozzarella, albahaca y aceite de oliva extra virgen',**

**@NuevoPrecio = 18.99,**

**@NuevaCategoria = 'Plato de Fondo'**

**GO**

**-- Ejecutar el procedimiento para buscar un plato por Nombre**

**EXEC spBuscarPlato**

**@Texto = 'Pizza',**

**@Criterio = 'Nombre'**

**GO**

**-- Ejecutar el procedimiento para buscar un plato por Categoria**

**EXEC spBuscarPlato**

**@Texto = 'Plato de Fondo',**

**@Criterio = 'Categoria'**

**GO**

**4.3Ejemplos para TUsuarios**

**--COMPROBACION**

**-- Ejecutar el procedimiento para listar todos los usuarios**

**EXEC spListarUsuario**

**GO**

**-- Ejecutar el procedimiento para agregar un usuario con los siguientes parámetros**

**EXEC spAgregarUsuario**

**@NombreUsuario = 'admin123',**

**@Clave = 'password123',**

**@TipoUsuario = 'Administrador'**

**GO**

**-- Ejecutar el procedimiento para eliminar un usuario con IdUsuario = 1**

**EXEC spEliminarUsuario @IdUsuario = 1**

**GO**

**-- Ejecutar el procedimiento para actualizar un usuario con IdUsuario = 1**

**EXEC spActualizarUsuario**

**@IdUsuario = 1,**

**@NuevoNombreUsuario = 'admin\_updated',**

**@NuevaClave = 'newpassword123',**

**@NuevoTipoUsuario = 'Chef'**

**GO**

**-- Ejecutar el procedimiento para buscar un usuario por IdUsuario**

**EXEC spBuscarUsuario**

**@Texto = '1',**

**@Criterio = 'IdUsuario'**

**GO**

**-- Ejecutar el procedimiento para buscar un usuario por NombreUsuario**

**EXEC spBuscarUsuario**

**@Texto = 'admin',**

**@Criterio = 'NombreUsuario'**

**GO**

**-- Ejecutar el procedimiento para buscar un usuario por TipoUsuario**

**EXEC spBuscarUsuario**

**@Texto = 'Chef',**

**@Criterio = 'TipoUsuario'**

**GO**

**4.4 Ejemplos para TClientes**

**--Procedimeintos xd**

**-- Ejecutar el procedimiento para listar todos los clientes**

**EXEC spListarCliente**

**GO**

**-- Ejecutar el procedimiento para agregar un cliente**

**EXEC spAgregarCliente**

**@IdCliente = 21,**

**@Nombre = 'Juan',**

**@Apellido = 'Perez',**

**@Telefono = '987654321',**

**@Correo = 'juanperez@email.com',**

**@Direccion = 'Calle Ficticia 123'**

**GO**

**-- Ejecutar el procedimiento para eliminar un cliente con IdCliente = 1**

**EXEC spEliminarCliente @IdCliente = 21**

**GO**

**-- Ejecutar el procedimiento para actualizar un cliente con IdCliente = 1**

**EXEC spActualizarCliente**

**@IdCliente = 1,**

**@NuevoNombre = 'Carlos',**

**@NuevoApellido = 'Gonzales',**

**@NuevoTelefono = '987654322',**

**@NuevoCorreo = 'carlosgonzales@email.com',**

**@NuevaDireccion = 'Avenida Central 456'**

**GO**

**-- Ejecutar el procedimiento para buscar un cliente por Nombre**

**EXEC spBuscarCliente**

**@Texto = 'Carlos',**

**@Criterio = 'Nombre'**

**GO**

**-- Ejecutar el procedimiento para buscar un cliente por Apellido**

**EXEC spBuscarCliente**

**@Texto = 'Perez',**

**@Criterio = 'Apellido'**

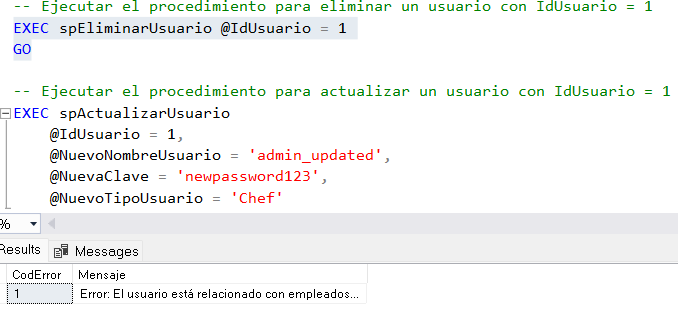
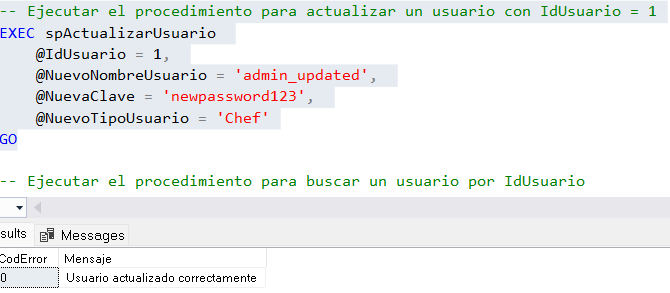
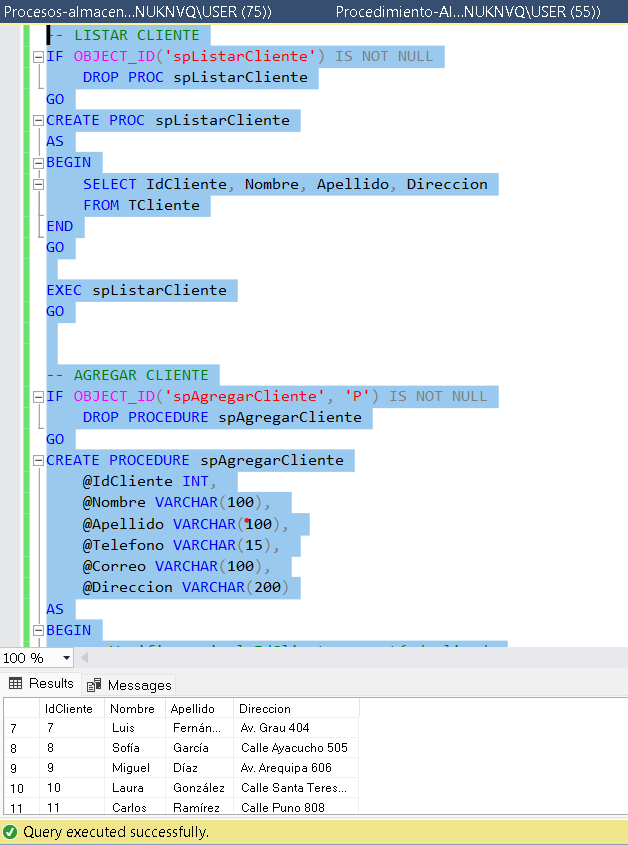
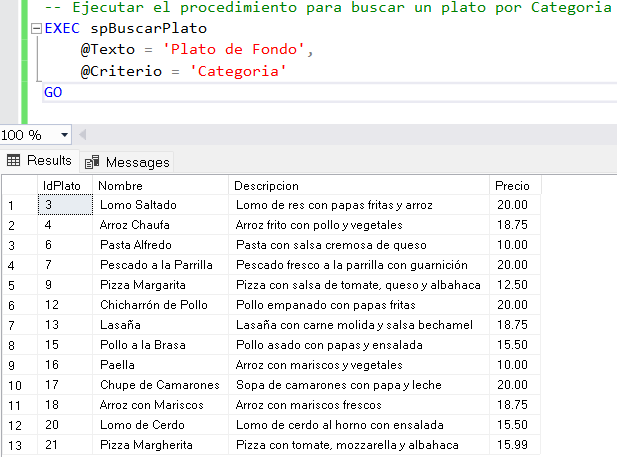
**GO**

1. **Anexos**
   1. **Scripts SQL**

[**https://github.com/AXXELYSS/PROCEDIMIENTOS**](https://github.com/AXXELYSS/PROCEDIMIENTOS)

* 1. **Acceso al Repositorio GitHub**

[**https://github.com/AXXELYSS/PROCEDIMIENTOS**](https://github.com/AXXELYSS/PROCEDIMIENTOS)

* 1. **Capturas o Evidencias de Funcionamiento**

1. **Presentación del Proyecto**
   1. **Descripción General**

### Descripción General del Proyecto:

Este sistema de base de datos tiene como objetivo centralizar la gestión de todos los aspectos operativos de un restaurante, facilitando las tareas tanto de los empleados como de los clientes. La plataforma se organiza en varias tablas interrelacionadas que permiten llevar un registro eficiente de los usuarios, clientes, empleados, mesas, reservas, pedidos y platos, con el fin de asegurar una experiencia fluida y eficiente.

* 1. **Demostración Técnica**

En la exposicion

* 1. **Preguntas y Respuestas**

1. **Conclusiones y Aprendizajes**

Gestión Eficiente: La base de datos organiza y facilita la gestión de clientes, empleados, reservas, pedidos y platos, mejorando la eficiencia operativa.

Integridad de los Datos: Las restricciones como claves primarias, foráneas y validaciones aseguran la coherencia y calidad de la información.

Flexibilidad: El sistema permite adaptarse a nuevas necesidades, como agregar más funcionalidades o categorías.

Mejora en el Servicio al Cliente: La automatización de reservas y pedidos optimiza tiempos y reduce errores.

Control de Roles: La gestión de usuarios con diferentes roles asegura un control adecuado sobre las funciones del sistema.

Visibilidad de Pedidos: Registrar detalles de los pedidos permite un mejor seguimiento y análisis de ventas.

Lecciones Técnicas: El buen diseño de la base de datos, el uso de SQL y las restricciones son fundamentales para crear sistemas confiables y escalables.