

Instituto Tecnológico de Costa Rica

Escuela de computación



Proyecto - Bases de Datos II:

Manual Técnico

Profesor: Alicia Salazar Hernandez

Elaborado por:

Roberto Calderon Jimenez

Jean Pierre Monterrey Arguedas

Israel Fabian Padilla Jenkins

Gabriel Omar Piedra Camacho

I SEMESTRE, 2019

1. Introduccion

1.1. Vision general

Este proyecto trata de una aplicación Web que permite la gestión de un sistema de ventas de automóviles que maneja desde la creación de los vehículos en fábrica hasta la distribución por sucursal de la compañía. Se podrá crear vehículos, administrar vehiculos, administrar sucursales, administrar fábrica y administrar empleados por parte del administrador, mientras que por gestión de clientes se tiene que pueden realizar diferentes consultas de vehículos, sucursales, empleados que trabajan en una sucursal, horarios y cercanía de la sucursal. También pueden realizar compra en línea de un vehículo.

El sistema posibilita las siguientes funciones:

- Inicio de Sesion (Administrador, Facturador y Cliente)
- Registro de nuevos clientes
- Vehículos: El administrador puede crear o borrar vehículos. Estos tienen diferentes características, como nombre, precio, año, tipo de combustible, tipo de vehículo, color, diferentes tipos de extras, puertas, cantidad de pasajeros e imagen que se pueden ver individualmente cuando se selecciona el vehículo en la aplicación Web. Además los vehículos están distribuidos a lo largo de las sucursales
- Sucursal: Las sucursales están repartidas a lo largo del país, un cliente puede ver y comprar los vehículos en stock de dicha sucursal además de la distancia del cliente a la sucursal, horarios de Trabajo y Personal del local. Además pueden recibir automóviles por consignación.
- Cuenta Administrador: Realiza el CRUD de vehiculos, ademas de reportes de Ventas por diferentes filtros, y reportes de vehículos más o menos vendidos
- Cuenta facturador: Un usuario facturador es el encargado de realizar la factura a cada cliente.
- Cuenta Cliente: Por su parte los clientes cuentan con las funcionalidades de consultar y comprar vehículos, ver datos de cada sucursal (distancia, empleados, horario de trabajo), además si el cliente es frecuente recibe descuentos especiales.

1.2. Definición del problema

El Problema consiste en la creación de una aplicación web para administrar el sistema de venta y compra de vehículos para cada uno de los tipos de usuarios, para que facilite y aumente las ganancias de la compañía CarroTec, y en el caso del cliente, un sistema fácil de usar y amigable que permita cumplir sus necesidades, en este caso la compra de vehículos de la compañía CarroTec.

1.3. Solución Propuesta

Una solución al problema brinda la posibilidad de promocionar la venta de vehículos de CarroTec por medio de una aplicación web que, para el administrador y cliente sean sencillas de utilizar y cumpla con los requisitos propuestos por parte de la Empresa. Se desarrollaron las siguientes partes:

- Base de Datos: La base de datos almacena todo tipo de información que será visualizada en la aplicación, desde los vehículos y sus características, sucursales, fabrica, sus eventos, tienda y noticias, como clientes y sus gustos por ciertas bandas a parte de la calificación que estos brindan a las bandas.

- Aplicación Web: La aplicación Web se encarga de comunicar a los usuarios con la base de datos. Esta permite al administrador realizar múltiples operaciones sobre las vehículos, sucursales y fábricas. Por su parte los clientes que poseen la funcionalidad de realizar consultas sobre las sucursales y los vehículos y la compra de vehículos. Y el facturador que se encarga de aprobar compra de vehículos a clientes físicos en la tienda y registrar nuevos clientes también.

Arriba se listan únicamente los requerimientos funcionales de la aplicación, pero adicionalmente es importante tomar en cuenta los siguientes requerimientos no funcionales:

- Velocidad de respuesta: Mientras se tenga una conexión estable a la base de datos se espera que el tiempo de respuesta sea bastante rápido. La información a transmitir no será de grandes volúmenes debido a que todas las operaciones son muy sencillas y de tamaño pequeño, por lo que cada operación debe durar una cantidad de tiempo muy pequeña.

- Velocidad de Navegación: el software cumple con un diseño sencillo por lo que la interacción con el usuario no debería ser un problema y cada operación realizada no debería tomar mucho tiempo.

1.4. Justificación

El objetivo de la empresa CarroTec es realizar más ventas en mas lugares y para más clientes, en otras palabras generar más ganancias. Por su parte los clientes quieren aplicaciones rápidas y sencillas de utilizar donde sus necesidades se cumplan a la perfección y sea 100% acertadas y correctas, por lo que realizar compras de vehículos y ver los datos de los mismos y las sucursales de CarroTec es necesario.

De esta forma CarroTec presenta una aplicación para que, tanto la empresa, como los clientes afiliados disfruten de una aplicación novedosa que cubra las necesidades de cada tipo de usuario mencionadas anteriormente.

1.5. Descripción del documento

Este documento está dirigido a cualquier persona con conocimientos básicos de elicitación de requerimientos, bases de datos y desarrollo de aplicaciones de software. Se aconseja una lectura natural del documento (de principio a fin, en ese orden) para entender cada apartado, porque se abordarán los temas de general a específico y si se avanza prematuramente a otra parte del documento se pueden perder detalles esenciales. A continuación se listan los siguientes apartados que documentan el desarrollo de la aplicación de CarroTec, una aplicación web que brinda las características antes mencionadas.

1.6. Tabla de Contenidos

Introduccion	1
Vision general	1
Definición del problema	2
Solución Propuesta	2
Justificacion	3
Descripción del documento	3
Tabla de Contenidos	4
Ámbitos del sistema	6
Objetivos	6
Alcances y suposiciones	6
Restricciones	6
Funcionamiento	6
Arquitectura del sistema	6
Descripción	6
Diagrama de Capas-paquetes	7
Diagrama de Despliegue	7
Diagrama del dominio de distribución de bases de Datos	8
Base de datos	8
Diagrama	8
Descripción de cada tabla	11
Restricciones de integridad referencial	11
Pruebas	11
Estrategia	11
Reflexión	11
Evaluación	11

Lecciones	11
Errores y limitaciones Conocidas	11
Apéndices	12
Scripts de SQL para la creación de tablas	12
Scripts de Procedimientos Almacenados	18
Scripts de Llenado de la base	69

2. Ámbitos del sistema

A Continuación se muestran fragmentos del código que cumplen con la descripción de los requerimientos.

2.1. Objetivos

Se detalla en la apéndice al final del documento el código de los procedimientos almacenados creados y utilizados. Destacan procedimientos almacenados como CRUD's para todas las tablas, Iniciar sesión, Facturar, Realizar pedido, realizar compra, Aprobar pedido, que son más específicos al funcionamiento de la aplicación.

2.2. Alcances y suposiciones

Para la simulación de la aplicación en funcionamiento se precargan datos en la base. También algunos elementos que pueden afectar el correcto funcionamiento de la aplicación puede ser la conexión del dispositivo a internet, si es muy lenta puede que no logre conectarse o la obtención de los datos sea ineficiente.

2.3. Restricciones

Debido a que es una aplicación para web, el usuario debe tener como mínimo un dispositivo de escritorio o móvil con conexión a internet estable.

2.4. Funcionamiento

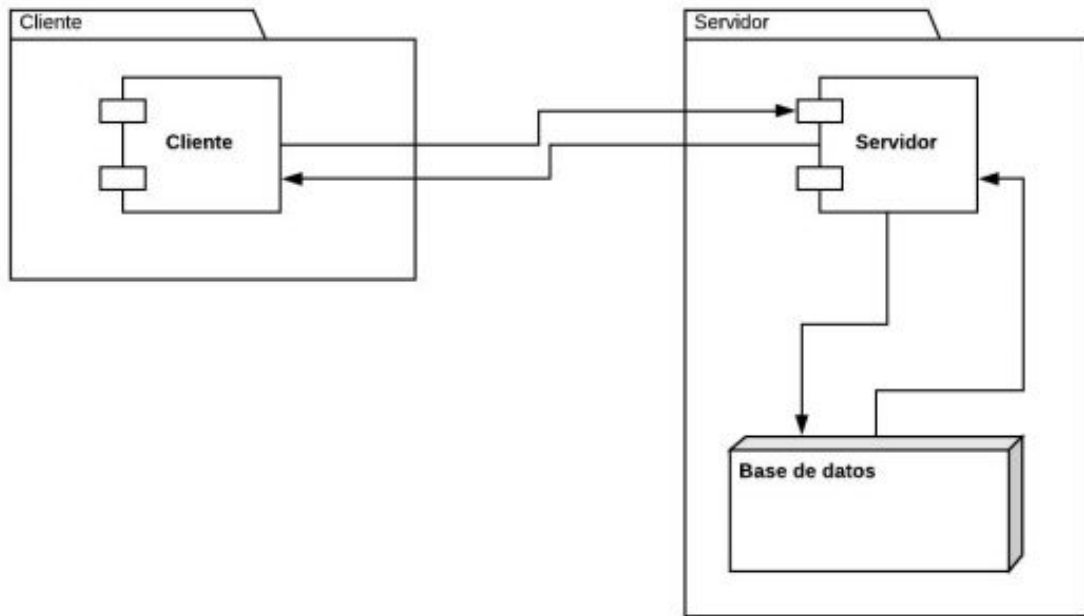
- Aplicación Web: JavaScript
- Base de datos: SQL Server 2016
- Conexión: Node.js

3. Arquitectura del sistema

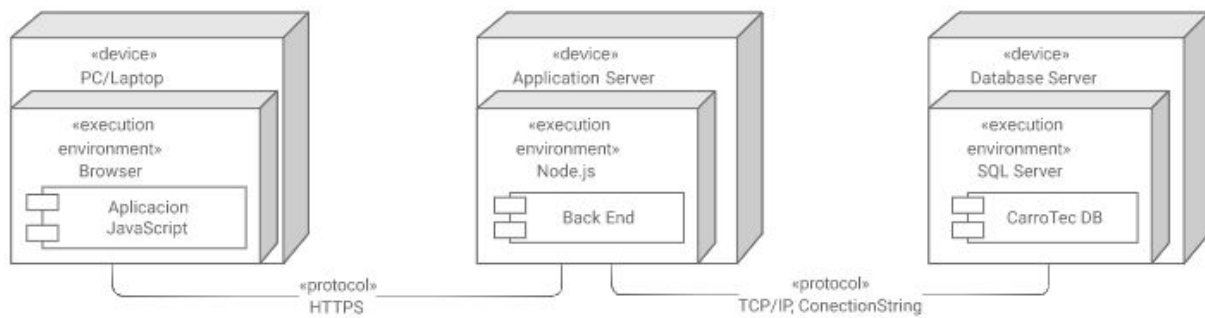
3.1. Descripción

Una base de datos sencilla que respeta las buenas prácticas de bases de datos apta para cumplir con los requerimientos del usuario por completo y eficientemente.

3.2. Diagrama de Capas-paquetes

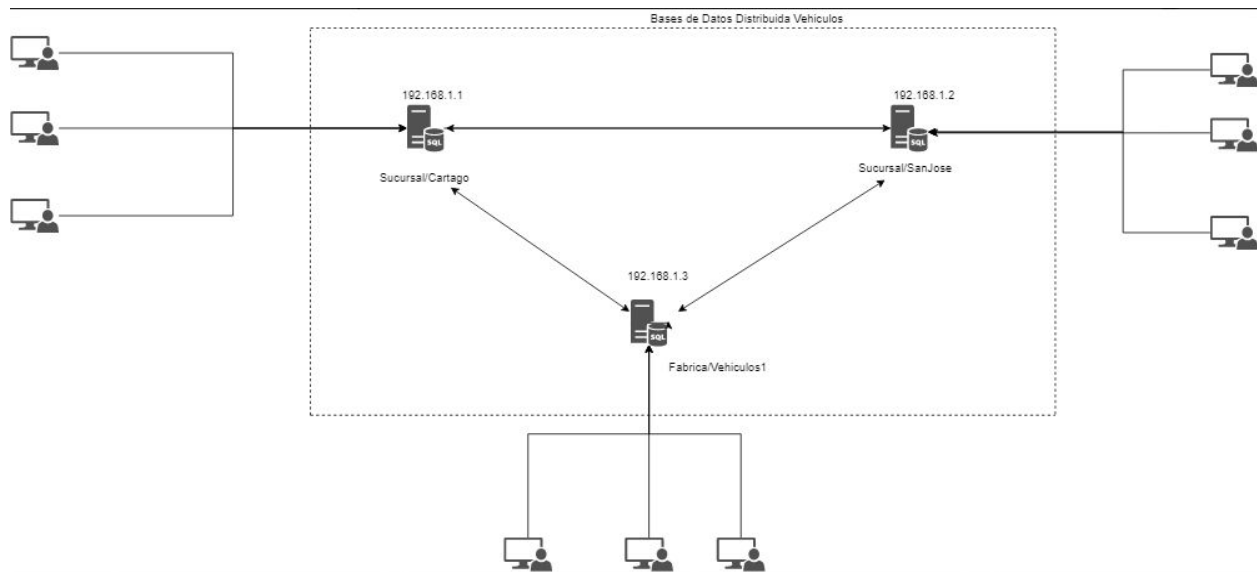


3.3. Diagrama de Despliegue



3.4. Diagrama del dominio de distribución de bases de Datos

Se presenta a continuación el diagrama de distribución de bases de Datos

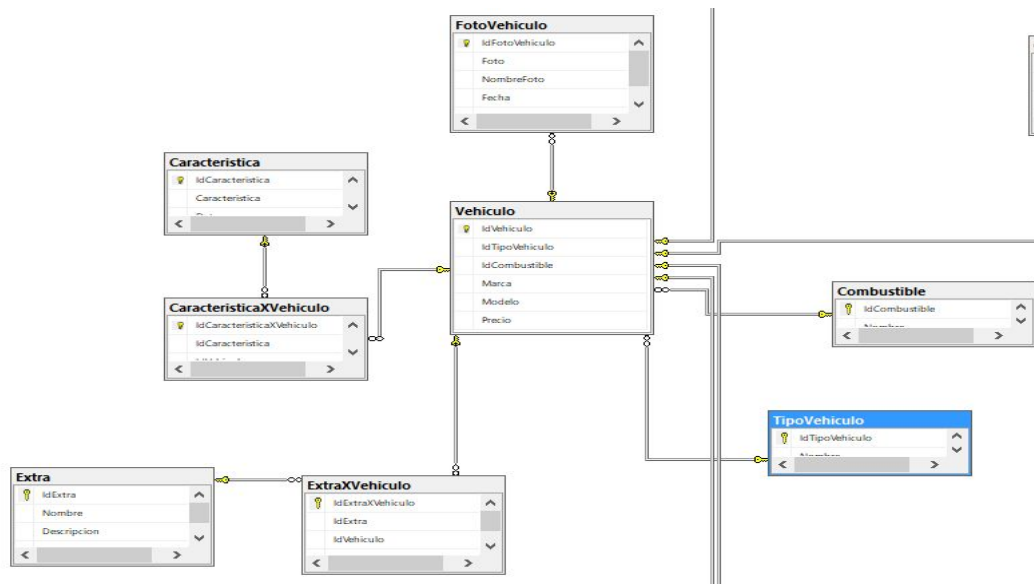


4. Base de datos

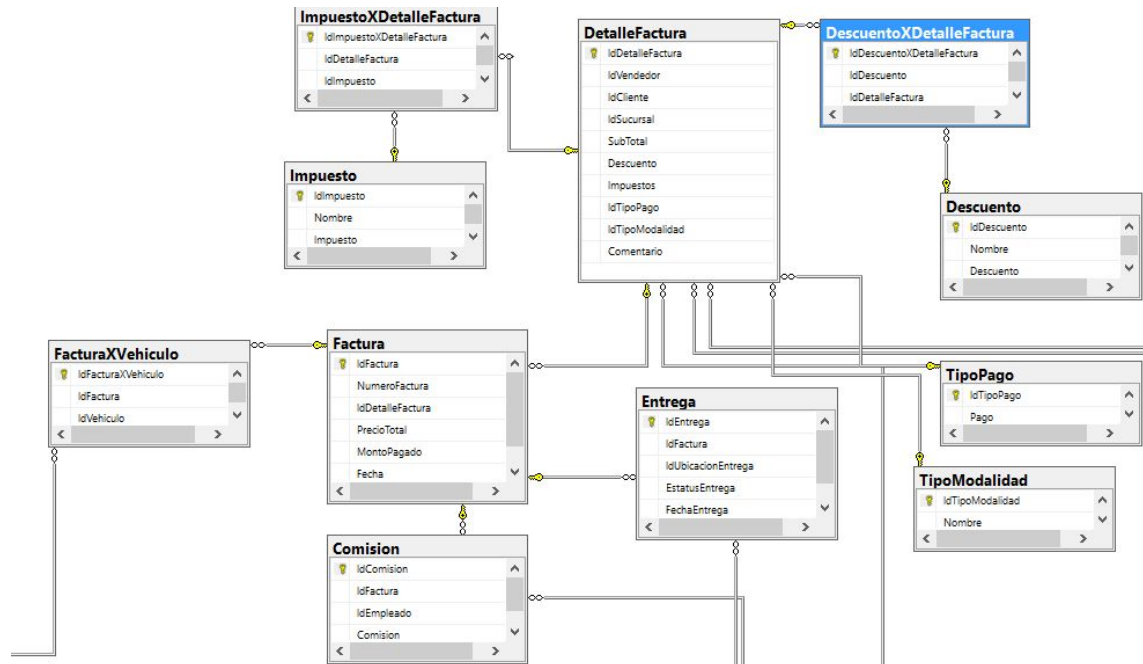
4.1. Diagrama

Se detalla el diagrama físico de la base de datos, se dividirá en secciones.

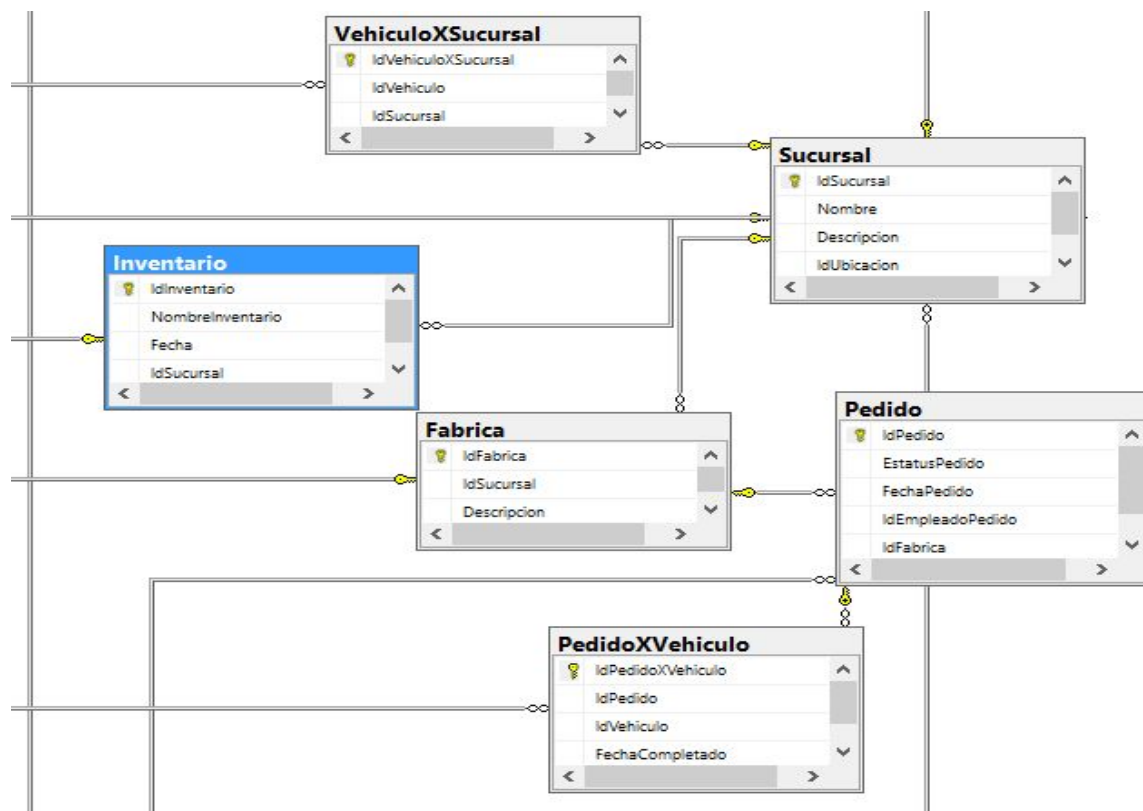
Seccion vehiculos:



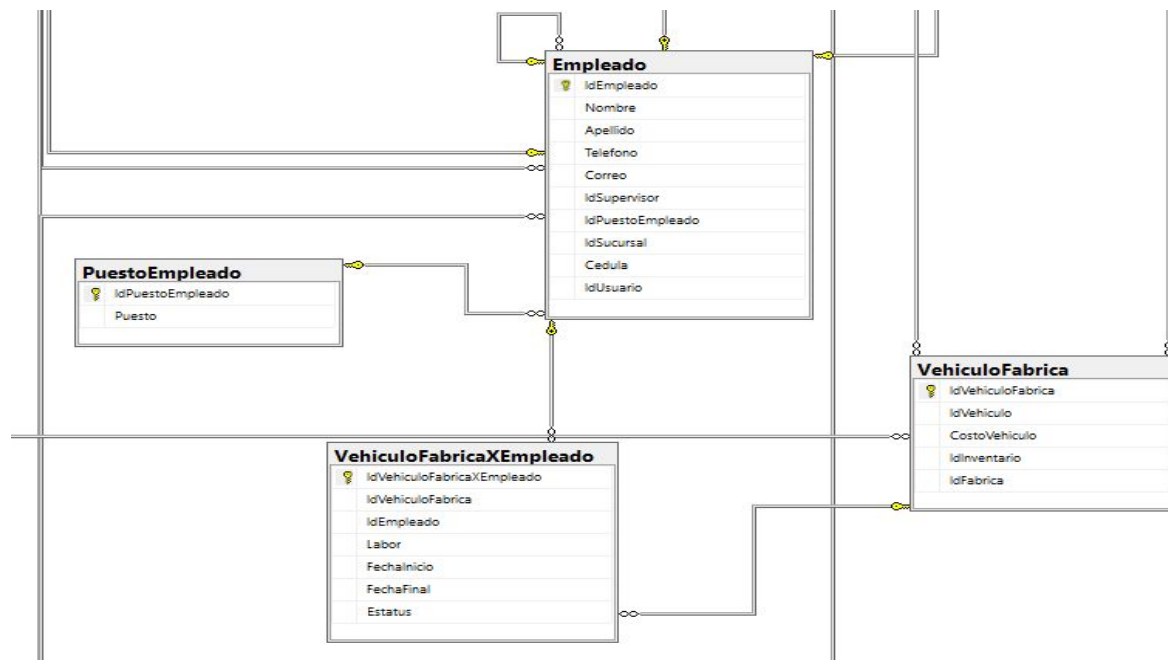
Sección Factura:



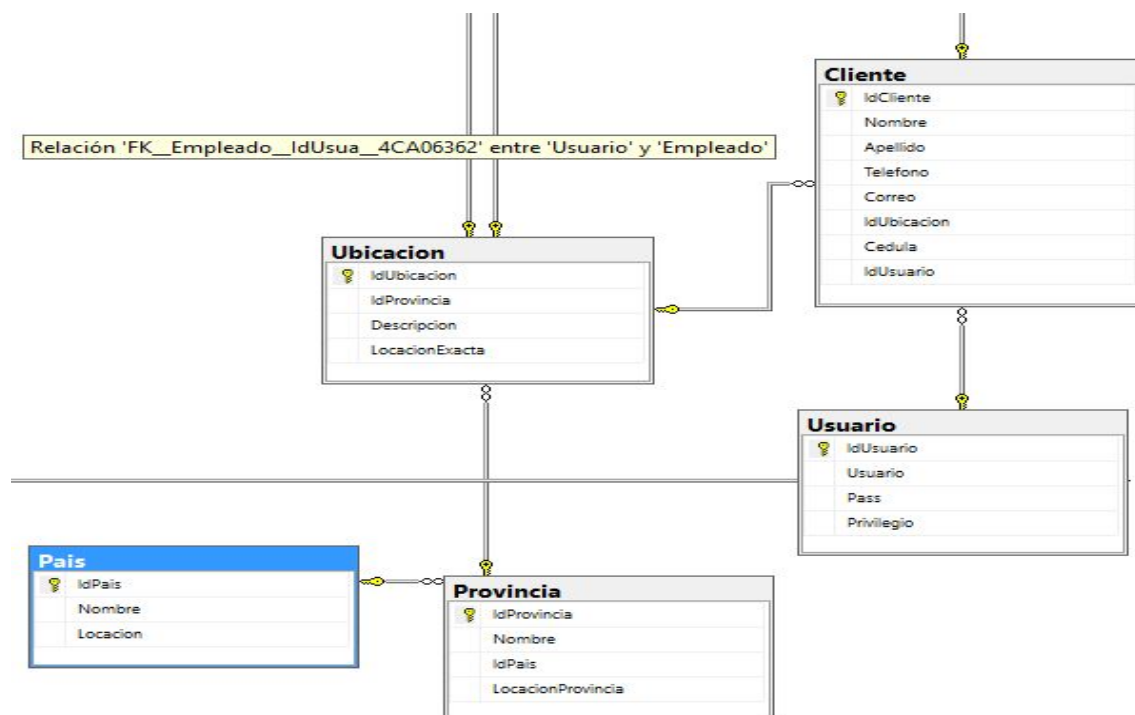
Sección Sucursal:



Sección Empleado:



Sección Cliente:



4.2. Descripción de cada tabla

Ver sección de apéndice, se detalla el código SQL usado para la creación de las tablas.

4.3. Restricciones de integridad referencial

Ver sección de apéndice, se detalla el código SQL usado para la creación de las tablas.

5. Pruebas

5.1. Estrategia

Ver sección de apéndice, se detalla el código SQL usado para rellenar la base, y a partir de ahí, realizar pruebas.

6. Reflexión

6.1. Evaluación

Aspectos importantes para el desarrollo de la aplicación:

Base de datos: Los integrantes están familiarizados con desarrollo y manipulación de bases de datos, sin embargo, la realización de una base de datos tan extensa es un trabajo duro, cansado y que consume mucho tiempo.

Conexión de la aplicación con la base de datos: Se invirtió tiempo mediano debido a que los integrantes ya han trabajado en este tipo de aplicaciones.

Interfaz: Se invirtió mucho tiempo porque la aplicación presenta muchas vistas.

6.2. Lecciones

Tiempo de desarrollo: El tiempo para realizar un proyecto de este tipo debe ser más de un mes, se contó con un tiempo de un mes justo para llevar a cabo todos los requerimientos el cual por la magnitud del proyecto trato de administrarse de la mejor manera posible pero siempre hay complicaciones especialmente en al fase final del proyecto, por lo que para futuros proyectos sería apreciado una mayor cantidad de tiempo.

6.3. Errores y limitaciones Conocidas

Tiempo de Desarrollo: El desarrollar un proyecto puede presentar una gran cantidad de inconvenientes , sin embargo el principal inconveniente en este caso es la lucha contra el tiempo , pues un error y su respectiva recuperación puede limitar por mucho el alcance del proyecto.

7. Apéndices

7.1. Scripts de SQL para la creación de tablas

```
CREATE DATABASE Proyecto_BasesII;  
GO  
USE Proyecto_BasesII
```

```
CREATE TABLE TipoModalidad(  
  IdTipoModalidad INT PRIMARY KEY IDENTITY,  
  Nombre VARCHAR(50)  
)
```

```
CREATE TABLE Pais(  
  IdPais INT PRIMARY KEY IDENTITY,  
  Nombre VARCHAR(50),  
  Locacion geometry  
)
```

```
CREATE TABLE Provincia(  
  IdProvincia INT PRIMARY KEY IDENTITY,  
  Nombre VARCHAR(50),  
  IdPais INT,  
  LocacionProvincia geometry,  
  FOREIGN KEY (IdPais) REFERENCES Pais(IdPais)  
)
```

```
CREATE TABLE Ubicacion(  
  IdUbicacion INT PRIMARY KEY IDENTITY,  
  IdProvincia INT,  
  Descripcion VARCHAR(50),  
  LocacionExacta geometry  
  FOREIGN KEY (IdProvincia) REFERENCES Provincia(IdProvincia)  
)
```

```
CREATE TABLE Caracteristica(  
  IdCaracteristica INT PRIMARY KEY IDENTITY,  
  Caracteristica VARCHAR(50),  
  Dato VARCHAR(50)  
)
```

```
CREATE TABLE Extra(  
  IdExtra INT PRIMARY KEY IDENTITY,  
  Nombre VARCHAR(50),  
  Descripcion VARCHAR(50),  
  Precio INT  
)
```

```
CREATE TABLE TipoVehiculo(  

```

```

    IdTipoVehiculo      INT PRIMARY KEY IDENTITY,
    Nombre              VARCHAR(50),
)

```

```

CREATE TABLE Combustible(
    IdCombustible        INT PRIMARY KEY IDENTITY,
    Nombre              VARCHAR(50)
)

```

```

CREATE TABLE Vehiculo(
    IdVehiculo           INT PRIMARY KEY IDENTITY,
    IdTipoVehiculo       INT,
    IdCombustible         INT,
    Marca                VARCHAR(50),
    Modelo               VARCHAR(50),
    Precio               INT,
    FOREIGN KEY (IdTipoVehiculo) REFERENCES TipoVehiculo(IdTipoVehiculo),
    FOREIGN KEY (IdCombustible) REFERENCES Combustible(IdCombustible)
)

```

```

CREATE TABLE FotoVehiculo(
    IdFotoVehiculo       INT PRIMARY KEY IDENTITY,
    Foto                 VARBINARY(MAX),
    NombreFoto           VARCHAR(50),
    Fecha                DATE,
    IdVehiculo           INT,
    FOREIGN KEY (IdVehiculo) REFERENCES Vehiculo(IdVehiculo)
)

```

```

CREATE TABLE ExtraXVehiculo(
    IdExtraXVehiculo     INT PRIMARY KEY IDENTITY,
    IdExtra               INT,
    IdVehiculo           INT,
    PrecioPagado          INT,
    FOREIGN KEY (IdExtra) REFERENCES Extra(IdExtra),
    FOREIGN KEY (IdVehiculo) REFERENCES Vehiculo(IdVehiculo)
)

```

```

CREATE TABLE CaracteristicaXVehiculo(
    IdCaracteristicaXVehiculo INT PRIMARY KEY IDENTITY,
    IdCaracteristica      INT,
    IdVehiculo            INT,
    FOREIGN KEY (IdCaracteristica) REFERENCES Caracteristica(IdCaracteristica),
    FOREIGN KEY (IdVehiculo) REFERENCES Vehiculo(IdVehiculo)
)

```

```

CREATE TABLE Sucursal(
    IdSucursal           INT PRIMARY KEY IDENTITY,

```

```

Nombre          VARCHAR(50),
Descripcion      VARCHAR(50),
IdUbicacion      INT,
FOREIGN KEY (IdUbicacion) REFERENCES Ubicacion(IdUbicacion)
)

```

```

CREATE TABLE PuestoEmpleado(
  IdPuestoEmpleado  INT PRIMARY KEY IDENTITY,
  Puesto            VARCHAR(50)
)

```

```

CREATE TABLE Usuario(
  IdUsuario  INT PRIMARY KEY IDENTITY,
  Usuario    VARCHAR(50),
  Pass       VARCHAR(50),
  Privilegio VARCHAR(50)
)

```

```

CREATE TABLE Empleado(
  IdEmpleado  INT PRIMARY KEY IDENTITY,
  Nombre      VARCHAR(50),
  Apellido    VARCHAR(50),
  Telefono    VARCHAR(50),
  Correo      VARCHAR(50),
  IdSupervisor  INT,
  IdPuestoEmpleado  INT,
  IdSucursal   INT,
  Cedula       VARCHAR(50),
  IdUsuario    INT,
  FOREIGN KEY (IdSupervisor) REFERENCES Empleado(IdEmpleado),
  FOREIGN KEY (IdPuestoEmpleado) REFERENCES PuestoEmpleado(IdPuestoEmpleado),
  FOREIGN KEY (IdSucursal) REFERENCES Sucursal(IdSucursal),
  FOREIGN KEY (IdUsuario) REFERENCES Usuario(IdUsuario)
)

```

```

CREATE TABLE Cliente(
  IdCliente  INT PRIMARY KEY IDENTITY,
  Nombre     VARCHAR(50),
  Apellido   VARCHAR(50),
  Telefono   VARCHAR(50),
  Correo     VARCHAR(50),
  IdUbicacion  INT,
  Cedula      VARCHAR(50),
  IdUsuario   INT,
  FOREIGN KEY (IdUbicacion) REFERENCES Ubicacion(IdUbicacion),
  FOREIGN KEY (IdUsuario) REFERENCES Usuario(IdUsuario)
)

```

```
CREATE TABLE Descuento(
  IdDescuento          INT PRIMARY KEY IDENTITY,
  Nombre               VARCHAR(50),
  Descuento            INT
)
```

```
CREATE TABLE Impuesto(
  IdImpuesto           INT PRIMARY KEY IDENTITY,
  Nombre               VARCHAR(50),
  Impuesto             INT
)
```

```
CREATE TABLE TipoPago(
  IdTipoPago           INT PRIMARY KEY IDENTITY,
  Pago                 VARCHAR(50)
)
```

```
CREATE TABLE DetalleFactura(
  IdDetalleFactura     INT PRIMARY KEY IDENTITY,
  IdVendedor           INT,
  IdCliente            INT,
  IdSucursal           INT,
  SubTotal             INT,
  Descuento            INT,
  Impuestos            INT,
  IdTipoPago           INT,
  IdTipoModalidad      INT,
  Comentario           VARCHAR(50),
  FOREIGN KEY (IdTipoPago) REFERENCES TipoPago(IdTipoPago),
  FOREIGN KEY (IdVendedor) REFERENCES Empleado(IdEmpleado),
  FOREIGN KEY (IdCliente) REFERENCES Cliente(IdCliente),
  FOREIGN KEY (IdSucursal) REFERENCES Sucursal(IdSucursal),
  FOREIGN KEY (IdTipoModalidad) REFERENCES TipoModalidad(IdTipoModalidad)
)
```

```
CREATE TABLE Factura(
  IdFactura            INT PRIMARY KEY IDENTITY,
  NumeroFactura        VARCHAR(50),
  IdDetalleFactura     INT,
  PrecioTotal          INT,
  MontoPagado          INT,
  Fecha               DATE
  FOREIGN KEY (IdDetalleFactura) REFERENCES DetalleFactura(IdDetalleFactura)
)
```

```
CREATE TABLE DescuentoXDetalleFactura(
  IdDescuentoXDetalleFactura INT PRIMARY KEY IDENTITY,
  IdDescuento          INT,
```



```

    IdDetalleFactura          INT
    FOREIGN KEY (IdDescuento) REFERENCES Descuento(IdDescuento),
    FOREIGN KEY (IdDetalleFactura) REFERENCES DetalleFactura(IdDetalleFactura)
)

CREATE TABLE ImpuestoXDetalleFactura(
    IdImpuestoXDetalleFactura  INT PRIMARY KEY IDENTITY,
    IdDetalleFactura          INT,
    IdImpuesto                INT
    FOREIGN KEY (IdDetalleFactura) REFERENCES DetalleFactura(IdDetalleFactura),
    FOREIGN KEY (IdImpuesto)   REFERENCES Impuesto(IdImpuesto)
)

CREATE TABLE FacturaXVehiculo(
    IdFacturaXVehiculo        INT PRIMARY KEY IDENTITY,
    IdFactura                 INT,
    IdVehiculo                INT,
    FOREIGN KEY (IdFactura)   REFERENCES Factura(IdFactura),
    FOREIGN KEY (IdVehiculo)  REFERENCES Vehiculo(IdVehiculo)
)

CREATE TABLE Comision(
    IdComision INT PRIMARY KEY IDENTITY,
    IdFactura  INT,
    IdEmpleado INT,
    Comision  INT
    FOREIGN KEY (IdFactura) REFERENCES Factura(IdFactura),
    FOREIGN KEY (IdEmpleado) REFERENCES Empleado(IdEmpleado)
)

CREATE TABLE Entrega(
    IdEntrega          INT PRIMARY KEY IDENTITY,
    IdFactura          INT,
    IdUbicacionEntrega INT,
    EstatusEntrega     VARCHAR(50),
    FechaEntrega       DATE,
    FOREIGN KEY (IdFactura) REFERENCES Factura(IdFactura),
    FOREIGN KEY (IdUbicacionEntrega) REFERENCES Ubicacion(IdUbicacion)
)

CREATE TABLE Fabrica(
    IdFabrica          INT PRIMARY KEY IDENTITY,
    IdSucursal         INT,
    Descripcion        VARCHAR(50),
    FOREIGN KEY (IdSucursal) REFERENCES Sucursal(IdSucursal)
)

CREATE TABLE Pedido(

```

```

IdPedido          INT PRIMARY KEY IDENTITY,
EstatusPedido     VARCHAR(50),
FechaPedido       DATE,
IdEmpleadoPedido  INT,
IdFabrica         INT
FOREIGN KEY (IdEmpleadoPedido) REFERENCES Empleado(IdEmpleado),
FOREIGN KEY (IdFabrica) REFERENCES Fabrica(IdFabrica)
)

CREATE TABLE PedidoXVehiculo(
IdPedidoXVehiculo INT PRIMARY KEY IDENTITY,
IdPedido          INT,
IdVehiculo        INT,
FechaCompletado   DATE
FOREIGN KEY (IdPedido) REFERENCES Pedido(IdPedido),
FOREIGN KEY (IdVehiculo) REFERENCES Vehiculo(IdVehiculo)
)

CREATE TABLE Inventario(
IdInventario      INT PRIMARY KEY IDENTITY,
NombreInventario  VARCHAR(50),
Fecha            DATE,
IdSucursal        INT,
FOREIGN KEY (IdSucursal) REFERENCES Sucursal(IdSucursal)
)

CREATE TABLE VehiculoFabrica(
IdVehiculoFabrica INT PRIMARY KEY IDENTITY,
IdVehiculo        INT,
CostoVehiculo     INT,
IdInventario      INT,
IdFabrica         INT,
FOREIGN KEY (IdFabrica) REFERENCES Fabrica(IdFabrica),
FOREIGN KEY (IdVehiculo) REFERENCES Vehiculo(IdVehiculo),
FOREIGN KEY (IdInventario) REFERENCES Inventario(IdInventario)
)

CREATE TABLE VehiculoFabricaXEmpleado(
IdVehiculoFabricaXEmpleado INT PRIMARY KEY IDENTITY,
IdVehiculoFabrica         INT,
IdEmpleado               INT,
Labor                   VARCHAR(50),
FechaInicio             DATE,
FechaFinal              DATE,
Estatus                 VARCHAR(50)
FOREIGN KEY (IdVehiculoFabrica) REFERENCES VehiculoFabrica(IdVehiculoFabrica),
FOREIGN KEY (IdEmpleado) REFERENCES Empleado(IdEmpleado)
)

```

```

CREATE TABLE VehiculoXSucursal(
    IdVehiculoXSucursal INT PRIMARY KEY IDENTITY,
    IdVehiculo          INT,
    IdSucursal          INT,
    FOREIGN KEY (IdVehiculo) REFERENCES Vehiculo(IdVehiculo),
    FOREIGN KEY (IdSucursal) REFERENCES Sucursal(IdSucursal)
)

```

/**Fin de Scripts de creación de tablas**/

7.2. Scripts de Procedimientos Almacenados

```

CREATE PROCEDURE [dbo].[ActualizarCliente](@IdCliente INT, @Nombre VARCHAR(50), @Apellido
VARCHAR(50), @Telefono VARCHAR(50), @Correo VARCHAR(50), @IdUbicacion INT)
AS
    IF @IdCliente IS NULL
        BEGIN
            RAISERROR(N'El ID es nulo', 1, 1);
        END
    IF @Telefono IS NOT NULL
        BEGIN
            DECLARE @ComprobarTelefono INT;
            SET @ComprobarTelefono = (SELECT IdCliente FROM Cliente WHERE IdCliente !=
@IdCliente AND Telefono = @Telefono)
            IF @ComprobarTelefono IS NOT NULL
                BEGIN
                    RAISERROR(N'Telefono utilizado', 1, 1);
                END
            END
    IF @Correo IS NOT NULL
        BEGIN
            DECLARE @ComprobarCorreo INT;
            SET @ComprobarCorreo = (SELECT IdCliente FROM Cliente WHERE IdCliente !=
@IdCliente AND Correo = @Correo)
            IF @ComprobarCorreo IS NOT NULL
                BEGIN
                    RAISERROR(N'Correo utilizado', 1, 1);
                END
            END
    IF @IdUbicacion IS NOT NULL
        BEGIN
            DECLARE @ComprobarUbicacion INT;
            SET @ComprobarUbicacion = (SELECT IdUbicacion FROM Ubicacion WHERE
IdUbicacion = @IdUbicacion)
            IF @ComprobarUbicacion IS NULL
                BEGIN
                    RAISERROR(N'La ubicacion no existe', 1, 1);
                END
            END
    END

```

```

BEGIN TRANSACTION
    UPDATE Cliente
    SET Nombre = @Nombre, Apellido = @Apellido, Telefono = @Telefono, Correo = @Correo,
    IdUbicacion = @IdUbicacion
    WHERE IdCliente = @IdCliente
COMMIT TRANSACTION
RETURN @@IDENTITY;
GO

```

```

CREATE PROCEDURE [dbo].[ActualizarComision](@IdComision INT, @IdFactura INT, @IdEmpleado INT,
@Comision INT)
AS
    IF @IdComision IS NULL
        RAISERROR(N'El ID es nulo', 1, 1);
    IF @IdFactura IS NOT NULL
        DECLARE @ComprobarFactura INT;
        SET @ComprobarFactura = (SELECT IdFactura FROM Factura WHERE IdFactura =
@IdFactura);
        IF @ComprobarFactura IS NULL
            RAISERROR(N'No existe la factura', 1, 1);
    IF @IdEmpleado IS NOT NULL
        DECLARE @ComprobarEmpleado INT;
        SET @ComprobarEmpleado = (SELECT IdEmpleado FROM Empleado WHERE IdEmpleado =
@IdEmpleado);
        IF @ComprobarEmpleado IS NULL
            RAISERROR(N'No existe el Empleado', 1, 1);
    BEGIN TRANSACTION
        UPDATE Comision
        SET IdFactura = ISNULL(@IdFactura, IdFactura), IdEmpleado = ISNULL(@IdEmpleado,
IdEmpleado), Comision = ISNULL(@Comision, Comision)
        WHERE IdComision = @IdComision
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO

```

```

CREATE PROCEDURE [dbo].[ActualizarDescuento](@IdDescuento INT, @Nombre VARCHAR(50), @Descuento
INT)
AS
    IF @IdDescuento IS NULL
        BEGIN
            RAISERROR(N'El ID es nulo', 1, 1);
        END
    BEGIN TRANSACTION
        UPDATE Descuento
        SET Nombre = ISNULL(@Nombre, Nombre), Descuento = ISNULL(@Descuento, Descuento)
        WHERE IdDescuento = @IdDescuento
    COMMIT TRANSACTION
    RETURN @@IDENTITY;

```

GO

```
CREATE PROCEDURE [dbo].[ActualizarDescuentoXDetalleFactura](@IdDescuentoXDetalleFactura INT,
@IdDescuento INT, @IdDetalleFactura INT)
AS
    IF @IdDescuentoXDetalleFactura IS NULL
        RAISERROR(N'El ID es nulo', 1, 1);
    IF @IdDescuento IS NOT NULL
        DECLARE @ComprobarDescuento INT;
        SELECT @ComprobarDescuento = (SELECT IdDescuento FROM Descuento WHERE
IdDescuento = @IdDescuento);
        IF @ComprobarDescuento IS NULL
            RAISERROR(N'El descuento no existe', 1, 1)
    IF @IdDetalleFactura IS NOT NULL
        DECLARE @ComprobarDetalleFactura INT;
        SELECT @ComprobarDetalleFactura = (SELECT IdDetalleFactura FROM DetalleFactura WHERE
IdDetalleFactura = @IdDetalleFactura);
        IF @ComprobarDetalleFactura IS NULL
            RAISERROR(N'El DetalleFactura no existe', 1, 1)
    BEGIN TRANSACTION
        UPDATE DescuentoXDetalleFactura
        SET IdDescuento = ISNULL(@IdDescuento, IdDescuento), IdDetalleFactura =
ISNULL(@IdDetalleFactura, IdDetalleFactura)
        WHERE IdDescuentoXDetalleFactura = @IdDescuentoXDetalleFactura
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO
```

```
CREATE PROCEDURE [dbo].[ActualizarDetalleFactura](@IdDetalleFactura INT, @IdVendedor INT, @IdCliente
INT, @SubTotal INT, @IdTipoPago INT, @Comentario VARCHAR(50))
AS
    IF @IdDetalleFactura IS NULL
        RAISERROR(N'El ID es nulo', 1, 1);
    IF @IdVendedor IS NOT NULL
        DECLARE @ComprobarVendedor INT;
        SET @ComprobarVendedor = (SELECT IdEmpleado FROM Empleado WHERE IdEmpleado =
@IdVendedor);
        IF @ComprobarVendedor IS NULL
            RAISERROR(N'No existe el vendedor', 1, 1);
    IF @IdCliente IS NOT NULL
        DECLARE @ComprobarCliente INT;
        SET @ComprobarCliente = (SELECT IdCliente FROM Cliente WHERE IdCliente =
@IdCliente);
        IF @ComprobarCliente IS NULL
            RAISERROR(N'No existe el Cliente', 1, 1);
    IF @IdTipoPago IS NOT NULL
        DECLARE @ComprobarTipoPago INT;
        SET @ComprobarTipoPago = (SELECT IdTipoPago FROM TipoPago WHERE IdTipoPago =
@IdTipoPago);
```

```

        IF @ComprobarTipoPago IS NULL
            RAISERROR(N'No existe el Tipo de Pago', 1, 1);
    BEGIN TRANSACTION
        UPDATE DetalleFactura
        SET IdVendedor = ISNULL(@IdVendedor, IdVendedor), IdCliente = ISNULL(@IdCliente,
        IdCliente), SubTotal = ISNULL(@SubTotal, SubTotal), IdTipoPago = ISNULL(@IdTipoPago, IdTipoPago),
            Comentario = ISNULL(@Comentario, Comentario)
        WHERE IdDetalleFactura = @IdDetalleFactura
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO

CREATE PROCEDURE [dbo].[ActualizarEntrega](@IdEntrega INT, @IdFactura INT, @IdUbicacionEntrega INT,
@EstatusEntrega VARCHAR(50), @FechaEntrega DATE)
AS
    IF @IdEntrega IS NULL
        RAISERROR(N'El ID es nulo', 1, 1);
    IF @IdFactura IS NOT NULL
        DECLARE @ComprobarFactura INT;
        SELECT @ComprobarFactura = (SELECT IdFactura FROM Factura WHERE IdFactura =
        @IdFactura);
        IF @ComprobarFactura IS NULL
            RAISERROR(N'No existe la factura', 1, 1)
    IF @IdUbicacionEntrega IS NOT NULL
        DECLARE @ComprobarUbicacionEntrega INT;
        SELECT @ComprobarUbicacionEntrega = (SELECT IdUbicacion FROM Ubicacion WHERE
        IdUbicacion = @IdUbicacionEntrega);
        IF @ComprobarUbicacionEntrega IS NULL
            RAISERROR(N'No existe la ubicacion', 1, 1)
    BEGIN TRANSACTION
        UPDATE Entrega
        SET IdFactura = ISNULL(@IdFactura, IdFactura), IdUbicacionEntrega =
        ISNULL(@IdUbicacionEntrega, IdUbicacionEntrega), EstatusEntrega = ISNULL(@EstatusEntrega, EstatusEntrega),
            FechaEntrega = ISNULL(@FechaEntrega, FechaEntrega)
        WHERE IdEntrega = @IdEntrega
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO

CREATE PROCEDURE [dbo].[ActualizarFactura](@IdFactura INT, @NumeroFactura VARCHAR(50),
@IdDetalleFactura INT, @PrecioTotal INT, @Fecha DATE)
AS
    IF @IdFactura IS NULL
        RAISERROR(N'El ID es nulo', 1, 1);
    IF @IdDetalleFactura IS NOT NULL
        DECLARE @ComprobarDetalle INT
        SET @ComprobarDetalle = (SELECT IdDetalleFactura FROM DetalleFactura WHERE
        IdDetalleFactura = @IdDetalleFactura)
        IF @ComprobarDetalle IS NULL
            RAISERROR(N'No existe el detalle de factura', 1, 1);

```

```

ELSE
    BEGIN
        BEGIN TRANSACTION
            UPDATE Factura
                SET NumeroFactura = ISNULL(@NumeroFactura, NumeroFactura),
IdDetalleFactura = ISNULL(@IdDetalleFactura, IdDetalleFactura), PrecioTotal = ISNULL(@PrecioTotal, PrecioTotal),
                Fecha = ISNULL(@Fecha, Fecha)
            WHERE IdFactura = @IdFactura
        COMMIT TRANSACTION
    END
GO

CREATE PROCEDURE [dbo].[ActualizarFacturaXVehiculo](@IdFacturaXVehiculo INT, @IdFactura INT,
@IdVehiculo INT)
AS
    IF @IdFacturaXVehiculo IS NULL
        RAISERROR(N'El ID es nulo', 1, 1);
    IF @IdFactura IS NOT NULL
        DECLARE @ComprobarFactura INT;
        SELECT @ComprobarFactura = (SELECT IdFactura FROM Factura WHERE IdFactura =
@IdFactura);
        IF @ComprobarFactura IS NULL
            RAISERROR(N'La Factura no existe', 1, 1)
    IF @IdVehiculo IS NOT NULL
        DECLARE @ComprobarVehiculo INT;
        SELECT @ComprobarVehiculo = (SELECT IdVehiculo FROM Vehiculo WHERE IdVehiculo =
@IdVehiculo);
        IF @ComprobarVehiculo IS NULL
            RAISERROR(N'El Vehiculo no existe', 1, 1)
    BEGIN TRANSACTION
        UPDATE FacturaXVehiculo
            SET IdFactura = ISNULL(@IdFactura, IdFactura), IdVehiculo = ISNULL(@IdVehiculo, IdVehiculo)
            WHERE IdFacturaXVehiculo = @IdFacturaXVehiculo
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO

CREATE PROCEDURE [dbo].[ActualizarImpuesto](@IdImpuesto INT, @Nombre VARCHAR(50), @Impuesto INT)
AS
    IF @IdImpuesto IS NULL
        BEGIN
            RAISERROR(N'El ID es nulo', 1, 1);
        END
    BEGIN TRANSACTION
        UPDATE Impuesto
            SET Nombre = ISNULL(@Nombre, Nombre), Impuesto = ISNULL(@Impuesto, Impuesto)
            WHERE IdImpuesto = @IdImpuesto
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO

```

```

CREATE PROCEDURE [dbo].[ActualizarImpuestoXDetalleFactura](@IdImpuestoXDetalleFactura INT,
@IdImpuesto INT, @IdDetalleFactura INT)
AS
    IF @IdImpuestoXDetalleFactura IS NULL
        RAISERROR(N'El ID es nulo', 1, 1);
    IF @IdImpuesto IS NOT NULL
        DECLARE @ComprobarImpuesto INT;
        SELECT @ComprobarImpuesto = (SELECT IdImpuesto FROM Impuesto WHERE IdImpuesto =
@IdImpuesto);
        IF @ComprobarImpuesto IS NULL
            RAISERROR(N'El Impuesto no existe', 1, 1)
    IF @IdDetalleFactura IS NOT NULL
        DECLARE @ComprobarDetalleFactura INT;
        SELECT @ComprobarDetalleFactura = (SELECT IdDetalleFactura FROM DetalleFactura WHERE
IdDetalleFactura = @IdDetalleFactura);
        IF @ComprobarDetalleFactura IS NULL
            RAISERROR(N'El DetalleFactura no existe', 1, 1)
    BEGIN TRANSACTION
        UPDATE ImpuestoXDetalleFactura
        SET IdImpuesto = ISNULL(@IdImpuesto, IdImpuesto), IdDetalleFactura =
ISNULL(@IdDetalleFactura, IdDetalleFactura)
        WHERE IdImpuestoXDetalleFactura = @IdImpuestoXDetalleFactura
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO

CREATE PROCEDURE [dbo].[ActualizarTipoPago](@IdTipoPago INT, @Pago VARCHAR(50))
AS
    IF @IdTipoPago IS NULL
        BEGIN
            RAISERROR(N'El ID es nulo', 1, 1);
        END
    BEGIN TRANSACTION
        UPDATE TipoPago
        SET Pago = ISNULL(@Pago, Pago)
        WHERE IdTipoPago = @IdTipoPago
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
GO

CREATE PROCEDURE [dbo].[AprobarPedido](@idPedido INT)
AS
    IF @idPedido IS NULL
        BEGIN
            RAISERROR(N'El id es necesario', 1, 1);
            RETURN;
        END
    DECLARE @comprobarPedido INT;
    SET
        @comprobarPedido = (SELECT idPedido FROM Pedido WHERE IdPedido = @idPedido)

```



```

IF @comprobarPedido IS NULL
    BEGIN
        RAISERROR(N'No existe el pedido', 1, 1);
        RETURN;
    END
BEGIN TRANSACTION
    UPDATE Pedido
    SET EstatusPedido = 'Aprobado'
    WHERE IdPedido = @idPedido
COMMIT TRANSACTION
RETURN @@IDENTITY;
GO

CREATE PROCEDURE [dbo].[Comprar](@idVehiculo INT, @idCliente INT, @comentario VARCHAR(50),
@impuesto INT, @descuento INT, @tipoPago VARCHAR(50), @tipoModalidad VARCHAR(50), @subTotal INT,
@idSucursal INT)
AS
    IF @idVehiculo IS NULL OR @idCliente IS NULL OR @comentario IS NULL OR @impuesto IS NULL OR
@descuento IS NULL OR @tipoPago IS NULL OR @tipoModalidad IS NULL
        BEGIN
            RAISERROR(N'Informacion incompleta', 1, 1);
            RETURN;
        END
    DECLARE @idTipoPago INT,
            @idTipoModalidad INT;
    SELECT @idTipoPago = (SELECT idTipoPago FROM TipoPago WHERE Pago = @tipoPago),
           @idTipoModalidad = (SELECT IdTipoModalidad FROM TipoModalidad WHERE Nombre
= @tipoModalidad);
    IF @idTipoPago IS NULL OR @idTipoModalidad IS NULL
        BEGIN
            RAISERROR(N'Informacion incorrecta', 1, 1);
            RETURN;
        END
    DECLARE @idDetalleFactura INT,
            @idFactura INT,
            @montoTotal INT;
    SET @montoTotal = @subTotal - (@subTotal * @descuento / 100);
    BEGIN TRANSACTION
        INSERT INTO DetalleFactura(IdCliente, IdSucursal, SubTotal, Descuento, Impuestos, IdTipoPago,
IdTipoModalidad, Comentario)
        VALUES(@idCliente, @idSucursal, @subTotal, @descuento, @impuesto, @idTipoPago,
@idTipoModalidad, @comentario)
        SET @idDetalleFactura = @@IDENTITY;
        INSERT INTO Factura(NumeroFactura, IdDetalleFactura, PrecioTotal, Fecha)
        VALUES(@idDetalleFactura, @idDetalleFactura, @montoTotal, GETDATE())
        SET @idFactura = @@IDENTITY;
        INSERT INTO FacturaXVehiculo(IdFactura, IdVehiculo)
        VALUES(@idFactura, @idVehiculo)
    COMMIT TRANSACTION
    RETURN @@IDENTITY;

```

GO

```
CREATE PROCEDURE [dbo].[Facturar](@idFactura INT, @idEmpleado INT)
AS
    IF @idFactura IS NULL OR @idEmpleado IS NULL
        BEGIN
            RAISERROR(N'Informacion incompleta', 1, 1)
            RETURN;
        END
    DECLARE @comprobarFactura INT,
            @comprobarEmpleado INT;
    SELECT @comprobarFactura = (SELECT IdDetalleFactura FROM Factura WHERE IdFactura =
@idFactura),
            @comprobarEmpleado = (SELECT idEmpleado FROM Empleado WHERE IdEmpleado =
@idEmpleado);
    IF @comprobarEmpleado IS NULL OR @comprobarFactura IS NULL
        BEGIN
            RAISERROR(N'Informacion incorrecta', 1, 1)
            RETURN;
        END
    BEGIN TRANSACTION
        UPDATE DetalleFactura
        SET IdVendedor = @idEmpleado
        WHERE IdDetalleFactura = @comprobarFactura
    COMMIT TRANSACTION
    RETURN @@IDENTITY;
```

GO

```
create procedure [dbo].[insertarCaracteristica]
    @Caracteristica varchar(50),
    @Dato varchar(50)
AS
BEGIN
    declare @DatoRepetido varchar(50)

    SELECT @DatoRepetido = Dato
    FROM Caracteristica
    WHERE Dato = @Dato

    IF @DatoRepetido IS NULL
        BEGIN
            BEGIN TRANSACTION
            insert into Caracteristica(Caracteristica,Dato)
            values (@Caracteristica,@Dato)
            COMMIT
            SELECT Caracteristica,Dato,IdCaracteristica
            FROM Caracteristica
        END
    ELSE
```

```

        BEGIN
        DECLARE @error varchar(50)
        SET      @error = 'La caracteristica ya existe'
        SELECT @error as Error
        END

END
GO

create procedure [dbo].[insertarCaracteristicaXVehiculo]
        @idCaracteristica int,
        @idVehiculo int

AS
BEGIN
    declare @VehiculoConfirmado int
    declare @CaracteristicaConfirmado int

    SELECT @VehiculoConfirmado = IdVehiculo
    FROM Vehiculo
    WHERE IdVehiculo = @idVehiculo

    SELECT @CaracteristicaConfirmado = IdCaracteristica
    FROM Caracteristica
    WHERE IdCaracteristica = @idCaracteristica

    IF @VehiculoConfirmado IS NOT NULL AND @CaracteristicaConfirmado IS NOT NULL
        BEGIN
            BEGIN TRANSACTION
            insert into CaracteristicaXVehiculo(IdCaracteristica,IdVehiculo)
            Values (@CaracteristicaConfirmado,@VehiculoConfirmado)
            COMMIT
            SELECT IdCaracteristica,IdVehiculo
            FROM CaracteristicaXVehiculo
            END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET      @error = 'El vehiculo o caracteristica indicado no existen'
            SELECT @error as Error
            END

END
GO

CREATE PROCEDURE [dbo].[InsertarCliente](@IdUsuario INT, @Nombre VARCHAR(50), @Apellido
VARCHAR(50), @Telefono VARCHAR(50), @Correo VARCHAR(50), @IdUbicacion INT)
AS
    IF @Nombre IS NULL OR @Apellido IS NULL OR @Telefono IS NULL OR @Correo IS NULL OR
@IdUbicacion IS NULL
        BEGIN
            RAISERROR(N'Informacion Incompleta', 1, 1);
        END

```

```

DECLARE @ComprobarCorreo INT,
        @ComprobarTelefono INT,
        @ComprobarUbicacion INT;
SELECT @ComprobarCorreo = (SELECT IdCliente FROM Cliente WHERE Correo = @Correo),
       @ComprobarTelefono = (SELECT IdCliente FROM Cliente WHERE Telefono =
@Telefono),
       @ComprobarUbicacion = (SELECT IdUbicacion FROM Ubicacion WHERE IdUbicacion =
@IdUbicacion);
IF @ComprobarCorreo IS NOT NULL
    BEGIN
        RAISERROR(N'Correo ya utilizado', 1, 1);
    END
ELSE IF @ComprobarTelefono IS NOT NULL
    BEGIN
        RAISERROR(N'Numero de telefono duplicado', 1, 1);
    END
ELSE IF @ComprobarUbicacion IS NULL
    BEGIN
        RAISERROR(N'No existe la ubicacion', 1, 1);
    END
ELSE
    BEGIN
        BEGIN TRANSACTION
        INSERT INTO Cliente(Nombre, Apellido, Telefono, Correo, IdUbicacion,
IdUsuario)
        VALUES(@Nombre, @Apellido, @Telefono, @Correo, @IdUbicacion,
@IdUsuario)
        COMMIT TRANSACTION
        RETURN @@IDENTITY;
    END
GO

create procedure [dbo].[insertarCombustible]
    @Nombre varchar(50)
AS
BEGIN
    declare @CombustibleRepetido varchar(50)

    SELECT @CombustibleRepetido = Nombre
    FROM Combustible
    WHERE Nombre = @Nombre

    IF @CombustibleRepetido IS NULL
        BEGIN
            BEGIN TRANSACTION
            insert into Combustible(Nombre)
            values (@Nombre)
            COMMIT
            SELECT Nombre, IdCombustible
            FROM Combustible

```

```

        END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El combustible ya existe'
            SELECT @error as Error
        END
    END
GO

CREATE PROCEDURE [dbo].[InsertarComision](@IdFactura INT, @IdEmpleado INT, @Comision INT)
AS
    IF @IdFactura IS NULL OR @IdEmpleado IS NULL OR @Comision IS NULL
        RAISERROR(N'Informacion incompleta', 1, 1);
    DECLARE @ComprobarFactura INT,
            @ComprobarEmpleado INT;
    SELECT @ComprobarFactura = (SELECT IdFactura FROM Factura WHERE IdFactura = @IdFactura),
           @ComprobarEmpleado = (SELECT IdEmpleado FROM Empleado WHERE IdEmpleado =
@IdEmpleado);
    IF @ComprobarFactura IS NULL
        RAISERROR(N'No existe la factura', 1, 1);
    ELSE IF @ComprobarEmpleado IS NULL
        RAISERROR(N'No existe el empleado', 1, 1);
    ELSE IF @Comision <= 0
        RAISERROR(N'Monto no valido', 1, 1);
    ELSE
        BEGIN
            BEGIN TRANSACTION
                INSERT INTO Comision(IdFactura, IdEmpleado, Comision)
                VALUES(@IdFactura, @IdEmpleado, @Comision)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
GO

CREATE PROCEDURE [dbo].[InsertarDescuento](@Nombre VARCHAR(50), @Descuento INT)
AS
    IF @Nombre IS NULL OR @Descuento IS NULL
        RAISERROR(N'Informacion Incompleta', 1, 1);
    ELSE
        BEGIN
            BEGIN TRANSACTION
                INSERT INTO Descuento(Nombre, Descuento)
                VALUES(@Nombre, @Descuento)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
GO

```

```

CREATE PROCEDURE [dbo].[InsertarDescuentoXDetalleFactura](@IdDescuento INT, @IdDetalleFactura INT)
AS
    IF @IdDescuento IS NULL OR @IdDetalleFactura IS NULL
        RAISERROR(N'Informacion incompleta', 1, 1);
    DECLARE @ComprobarDescuento INT,
            @ComprobarDetalle INT;
    SELECT @ComprobarDescuento = (SELECT IdDescuento FROM Descuento WHERE IdDescuento =
@IdDescuento),
            @ComprobarDetalle = (SELECT IdDetalleFactura FROM DetalleFactura WHERE
IdDetalleFactura = @IdDetalleFactura)
    IF @ComprobarDescuento IS NULL
        RAISERROR(N'El descuento no existe', 1, 1)
    ELSE IF @ComprobarDetalle IS NULL
        RAISERROR(N'El detalle no existe', 1, 1)
    ELSE
        BEGIN
            BEGIN TRANSACTION
                INSERT INTO DescuentoXDetalleFactura(IdDescuento, IdDetalleFactura)
                VALUES(@IdDescuento, @IdDetalleFactura)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
GO

CREATE PROCEDURE [dbo].[InsertarDetalleFactura](@IdVendedor INT, @IdCliente INT, @SubTotal INT,
@IdTipoPago INT, @Comentario VARCHAR(50), @TipoModalidad INT)
AS
    IF @IdVendedor IS NULL OR @IdCliente IS NULL OR @SubTotal IS NULL OR @IdTipoPago IS NULL OR
@Comentario IS NULL
        RAISERROR(N'Informacion Incompleta', 1, 1);
    DECLARE @ComprobarVendedor INT,
            @ComprobarCliente INT,
            @ComprobarTipoPago INT;
    SELECT @ComprobarVendedor = (SELECT IdEmpleado FROM Empleado WHERE IdEmpleado =
@IdVendedor),
            @ComprobarCliente = (SELECT IdCliente FROM Cliente WHERE IdCliente =
@IdCliente),
            @ComprobarTipoPago = (SELECT IdTipoPago FROM TipoPago WHERE IdTipoPago =
@IdTipoPago);
    IF @ComprobarVendedor IS NULL
        RAISERROR(N'No existe el vendedor', 1, 1);
    ELSE IF @ComprobarCliente IS NULL
        RAISERROR(N'No existe el cliente', 1, 1);
    ELSE IF @ComprobarTipoPago IS NULL
        RAISERROR(N'No existe el metodo de pago', 1, 1);
    ELSE
        BEGIN
            BEGIN TRANSACTION
                INSERT INTO DetalleFactura(IdVendedor, IdCliente, SubTotal, IdTipoPago,
Comentario, IdTipoModalidad)

```

```

VALUES(@IdVendedor, @IdCliente, @SubTotal, @IdTipoPago, @Comentario,
@TipoModalidad)

COMMIT TRANSACTION
RETURN @@IDENTITY;

END

GO

create procedure [dbo].[insertarEmpleadoFabrica]
    @Nombre varchar(50),
    @Apellido varchar(50),
    @Telefono varchar(50),
    @Correo varchar(50),
    @Supervisor varchar(50),
    @Puesto varchar(50),
    @Fabrica int,
    @Cedula varchar(50),
    @idUsuario int

AS
BEGIN
    declare @CedulaRepetida varchar(50)
    declare @FabricaConfirmada int
    declare @SupervisorConfirmado int
    declare @PuestoConfirmado int

    SELECT @CedulaRepetida = Cedula
    FROM Empleado
    WHERE Cedula = @Cedula

    SELECT @FabricaConfirmada = IdFabrica
    FROM Fabrica
    WHERE IdFabrica = @Fabrica

    SELECT @SupervisorConfirmado = IdEmpleado
    FROM Empleado
    WHERE Cedula = @Supervisor

    SELECT @PuestoConfirmado = IdPuestoEmpleado
    FROM PuestoEmpleado
    WHERE Puesto = @Puesto

    IF @CedulaRepetida IS NULL AND @FabricaConfirmada IS NOT NULL AND @PuestoConfirmado IS NOT
NULL
        BEGIN
            BEGIN TRANSACTION
            insert into
Empleado(Nombre,Apellido,Telefono,Correo,IdSupervisor,IdPuestoEmpleado,Cedula,IdUsuario)
values
(@Nombre,@Apellido,@Telefono,@Correo,@SupervisorConfirmado,@PuestoConfirmado,@Cedula,@idUsuario)
            insert into EmpleadoXFabrica(IdEmpleado,IdFabrica)

```

```

        values(@@IDENTITY,@FabricaConfirmada)
        COMMIT
        SELECT IdEmpleado,Nombre,Apellido,Telefono,Correo,IdSupervisor,IdPuestoEmpleado,Cedula
        FROM Empleado
        END
ELSE
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'La cedula indicada ya esta registrada o la sucursal o puesto no existen'
        SELECT @error as Error
        END
END
GO

create procedure [dbo].[insertarEmpleadoSucursal]
    @Nombre varchar(50),
        @Apellido varchar(50),
        @Telefono varchar(50),
        @Correo varchar(50),
        @Supervisor varchar(50),
        @Puesto varchar(50),
        @Sucursal int,
        @Cedula varchar(50),
        @idUsuario int

AS
BEGIN
    declare @CedulaRepetida varchar(50)
    declare @SucursalConfirmada int
    declare @SupervisorConfirmado int
    declare @PuestoConfirmado int

    SELECT @CedulaRepetida = Cedula
    FROM Empleado
    WHERE Cedula = @Cedula

    SELECT @SucursalConfirmada = IdSucursal
    FROM Sucursal
    WHERE IdSucursal = @Sucursal

    SELECT @SupervisorConfirmado = IdEmpleado
    FROM Empleado
    WHERE Cedula = @Supervisor

    SELECT @PuestoConfirmado = IdPuestoEmpleado
    FROM PuestoEmpleado
    WHERE Puesto = @Puesto

    IF @CedulaRepetida IS NULL AND @SucursalConfirmada IS NOT NULL AND @PuestoConfirmado IS
NOT NULL

```



```

        BEGIN
        BEGIN TRANSACTION
        insert into
Empleado(Nombre,Apellido,Telefono,Correo,IdSupervisor,IdPuestoEmpleado,Cedula,IdUsuario)
        values
(@Nombre,@Apellido,@Telefono,@Correo,@SupervisorConfirmado,@PuestoConfirmado,@Cedula,@idUsuario)
        insert into EmpleadoXSucursal(IdEmpleado,IdSucursal)
        values(@@IDENTITY,@SucursalConfirmada)
        COMMIT
        SELECT IdEmpleado,Nombre,Apellido,Telefono,Correo,IdSupervisor,IdPuestoEmpleado,Cedula
        FROM Empleado
        END
ELSE
        BEGIN
        DECLARE @error varchar(50)
        SET @error = 'La cedula indicada ya esta registrada o la sucursal o puesto no existen'
        SELECT @error as Error
        END
END
GO

CREATE PROCEDURE [dbo].[InsertarEntrega](@IdFactura INT, @IdUbicacionEntrega INT, @EstatusEntrega
VARCHAR(50), @FechaEntrega DATE)
AS
        IF @IdFactura IS NULL OR @IdUbicacionEntrega IS NULL OR @EstatusEntrega IS NULL OR
@FechaEntrega IS NULL
                RAISERROR(N'Informacion Incompleta', 1, 1);
        DECLARE @ComprobarFactura INT,
                @ComprobarUbicacion INT;
        SELECT @ComprobarFactura = (SELECT IdFactura FROM Factura WHERE IdFactura = @IdFactura),
                @ComprobarUbicacion = (SELECT IdUbicacion FROM Ubicacion WHERE IdUbicacion =
@IdUbicacionEntrega);
        IF @ComprobarFactura IS NULL
                RAISERROR(N'No existe la factura', 1, 1)
        ELSE IF @ComprobarUbicacion IS NULL
                RAISERROR(N'No existe la ubicacion', 1, 1)
        ELSE
                BEGIN
                BEGIN TRANSACTION
                INSERT INTO Entrega(IdFactura, IdUbicacionEntrega, EstatusEntrega,
FechaEntrega)
                VALUES(@IdFactura, @IdUbicacionEntrega, @EstatusEntrega,
@FechaEntrega)
                COMMIT TRANSACTION
                RETURN @@IDENTITY;
                END
        GO

create procedure [dbo].[insertarExtra]
        @Nombre varchar(50),

```

```

                                @Descripcion varchar(50),
                                @Precio int
AS
BEGIN
    declare @ExtraRepetido varchar(50)

    SELECT @ExtraRepetido = Nombre
    FROM Extra
    WHERE Nombre = @Nombre

    IF @ExtraRepetido IS NULL
        BEGIN
            BEGIN TRANSACTION
            insert into Extra(Nombre,Descripcion,Precio)
            values (@Nombre,@Descripcion,@Precio)
            COMMIT
            SELECT Nombre,Descripcion,Precio,IdExtra
            FROM Extra
        END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El extra ya existe'
            SELECT @error as Error
        END
END
GO

create procedure [dbo].[insertarExtraXVehiculo]
                                @idExtra int,
                                @idVehiculo int
AS
BEGIN
    declare @VehiculoConfirmado int
    declare @ExtraConfirmado int
    declare @Precio int

    SELECT @VehiculoConfirmado = IdVehiculo
    FROM Vehiculo
    WHERE IdVehiculo = @idVehiculo

    SELECT @ExtraConfirmado = IdExtra,@Precio = Precio
    FROM Extra
    WHERE IdExtra = @idExtra

    IF @VehiculoConfirmado IS NOT NULL AND @ExtraConfirmado IS NOT NULL
        BEGIN
            BEGIN TRANSACTION
            insert into ExtraXVehiculo(IdExtra,IdVehiculo,PrecioPagado)
            Values (@ExtraConfirmado,@VehiculoConfirmado,@Precio)

```

```

        COMMIT
        SELECT IdExtra,IdVehiculo,PrecioPagado
        FROM ExtraXVehiculo
    END
ELSE
    BEGIN
        DECLARE @error varchar(50)
        SET      @error = 'El vehiculo o extra indicado no existen'
        SELECT @error as Error
    END
END
GO

CREATE  PROCEDURE [dbo].[InsertarFabrica]
        @Descripcion varchar(50),
        @IdUbicacion  int
AS
BEGIN

    Begin Transaction;
    insert into Fabrica(IdUbicacion,Descripcion)
    values(@IdUbicacion,@Descripcion)
    commit
END
GO

CREATE  PROCEDURE [dbo].[InsertarFactura](@NumeroFactura VARCHAR(50), @IdDetalleFactura INT,
@PrecioTotal INT, @Fecha DATE)
AS
    IF @NumeroFactura IS NULL OR @IdDetalleFactura IS NULL OR @PrecioTotal IS NULL OR @Fecha IS
    NULL
        RAISERROR(N'Informacion Incompleta', 1, 1);
    DECLARE @ComprobarDetalle INT
    SET @ComprobarDetalle = (SELECT IdDetalleFactura FROM DetalleFactura WHERE IdDetalleFactura =
    @IdDetalleFactura)
    IF @ComprobarDetalle IS NULL
        RAISERROR(N'No existe el detalle de factura', 1, 1);
    ELSE
        BEGIN
            BEGIN TRANSACTION
            INSERT INTO Factura(NumeroFactura, IdDetalleFactura, PrecioTotal, Fecha)
            VALUES (@NumeroFactura, @IdDetalleFactura, @PrecioTotal, @Fecha)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
    END
GO

CREATE  PROCEDURE [dbo].[InsertarFacturaXVehiculo](@IdFactura INT, @IdVehiculo INT)
AS
    IF @IdFactura IS NULL OR @IdVehiculo IS NULL

```

```

        RAISERROR(N'Informacion incompleta', 1, 1);
    DECLARE @ComprobarFactura INT,
            @ComprobarDetalle INT;
    SELECT @ComprobarFactura = (SELECT IdFactura FROM Factura WHERE IdFactura = @IdFactura),
           @ComprobarDetalle = (SELECT IdVehiculo FROM Vehiculo WHERE IdVehiculo =
@IdVehiculo)
    IF @ComprobarFactura IS NULL
        RAISERROR(N'La Factura no existe', 1, 1)
    ELSE IF @ComprobarDetalle IS NULL
        RAISERROR(N'El vehiculo no existe', 1, 1)
    ELSE
        BEGIN
            BEGIN TRANSACTION
            INSERT INTO FacturaXVehiculo(IdFactura , IdVehiculo)
            VALUES(@IdFactura , @IdVehiculo)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
GO

create procedure [dbo].[insertarFotoVehiculo]
    @foto nvarchar(128),
    @Nombre varchar(50),
    @fecha date,
    @idVehiculo int
AS
BEGIN
    declare @VehiculoConfirmado int

    SELECT @VehiculoConfirmado = IdVehiculo
    FROM Vehiculo
    WHERE IdVehiculo = @idVehiculo

    IF @VehiculoConfirmado IS NOT NULL
        BEGIN
            BEGIN TRANSACTION
            INSERT INTO FotoVehiculo (Fecha,Foto,NombreFoto,IdVehiculo)
            VALUES (GETDATE(),@foto,@Nombre,@idVehiculo)
            COMMIT
            SELECT IdFotoVehiculo,Foto,NombreFoto,Fecha,IdVehiculo
            FROM FotoVehiculo
            END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El vehiculo indicado no existe'
            SELECT @error as Error
            END
END
GO

```

```

CREATE PROCEDURE [dbo].[InsertarImpuesto](@Nombre VARCHAR(50), @Impuesto INT)
AS
    IF @Nombre IS NULL OR @Impuesto IS NULL
        RAISERROR(N'Informacion Incompleta', 1, 1);
    ELSE
        BEGIN
            BEGIN TRANSACTION
                INSERT INTO Impuesto(Nombre, Impuesto)
                VALUES(@Nombre, @Impuesto)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
GO

CREATE PROCEDURE [dbo].[InsertarImpuestoXDetalleFactura](@IdImpuesto INT, @IdDetalleFactura INT)
AS
    IF @IdImpuesto IS NULL OR @IdDetalleFactura IS NULL
        RAISERROR(N'Informacion incompleta', 1, 1);
    DECLARE @ComprobarImpuesto INT,
            @ComprobarDetalle INT;
    SELECT @ComprobarImpuesto = (SELECT IdImpuesto FROM Impuesto WHERE IdImpuesto =
@IdImpuesto),
           @ComprobarDetalle = (SELECT IdDetalleFactura FROM DetalleFactura WHERE
IdDetalleFactura = @IdDetalleFactura)
    IF @ComprobarImpuesto IS NULL
        RAISERROR(N'El Impuesto no existe', 1, 1)
    ELSE IF @ComprobarDetalle IS NULL
        RAISERROR(N'El detalle no existe', 1, 1)
    ELSE
        BEGIN
            BEGIN TRANSACTION
                INSERT INTO ImpuestoXDetalleFactura(IdImpuesto, IdDetalleFactura)
                VALUES(@IdImpuesto, @IdDetalleFactura)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
GO

CREATE PROCEDURE [dbo].[InsertarInventario]
    @NombreInventario varchar(50), @Fecha date, @IdSucursal int
AS
BEGIN
    Begin Transaction;
    insert into Inventario(NombreInventario, Fecha, IdSucursal)
    values(@NombreInventario, @Fecha, @IdSucursal)
    commit
END
GO

```

```

create procedure [dbo].[insertarPais]
    @Nombre varchar(50),
    @Locacion geometry
AS
BEGIN
    declare @PaisRepetido varchar(50)
    SELECT @PaisRepetido = Nombre
    FROM Pais
    WHERE Nombre = @Nombre

    IF @PaisRepetido IS NULL
    BEGIN
        BEGIN TRANSACTION
        insert into Pais(Nombre,Locacion)
        values (@Nombre,@Locacion)
        COMMIT
        SELECT Nombre,Locacion,IdPais
        FROM Pais
    END
    ELSE
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'El pais ya existe en la base'
        SELECT @error as Error
    END
END
GO

CREATE PROCEDURE [dbo].[InsertarPedido]
    @EstatusPedido Varchar(50), @FechaPedido Date, @IdEmpleadoPedido int, @IdFabrica int
AS
BEGIN
    Begin Transaction;
    insert into Pedido(EstatusPedido,FechaPedido,IdEmpleadoPedido,IdFabrica)
    values(@EstatusPedido,@FechaPedido,@IdEmpleadoPedido,@IdFabrica)
    commit
END
GO

CREATE PROCEDURE [dbo].[InsertarPedidoXVehiculo]
    @IdPedido int, @IdVehiculo int, @FechaCompletado date
AS
BEGIN
    Begin Transaction;
    insert into PedidoXVehiculo(IdPedido,IdVehiculo,FechaCompletado)
    values(@IdPedido,@IdVehiculo,@FechaCompletado)
    commit
END
GO

```

```

create procedure [dbo].[insertarProvincia]
    @Nombre varchar(50),
    @Locacion geometry,
    @idPais int
AS
BEGIN
    declare @ProvinciaRepetida varchar(50)
    declare @PaisConfirmado int

    SELECT @ProvinciaRepetida = Nombre
    FROM Provincia
    WHERE Nombre = @Nombre

    SELECT @PaisConfirmado = IdPais
    FROM Pais
    WHERE IdPais = @idPais

    IF @ProvinciaRepetida IS NULL AND @PaisConfirmado IS NOT NULL
        BEGIN
            BEGIN TRANSACTION
            insert into Provincia(Nombre,IdPais,LocacionProvincia)
            values (@Nombre,@PaisConfirmado,@Locacion)
            COMMIT
            SELECT Nombre,LocacionProvincia,IdProvincia,IdPais
            FROM Provincia
            END
        ELSE
            BEGIN
                DECLARE @error varchar(50)
                SET @error = 'La provincia ya existe en la base'
                SELECT @error as Error
            END
END
GO

```

```

create procedure [dbo].[insertarPuestoEmpleado]
    @Puesto varchar(50)
AS
BEGIN
    declare @PuestoRepetido varchar(50)

    SELECT @PuestoRepetido = Puesto
    FROM PuestoEmpleado
    WHERE Puesto = @Puesto

    IF @PuestoRepetido IS NULL
        BEGIN
            BEGIN TRANSACTION
            insert into PuestoEmpleado(Puesto)

```

```

        values (@Puesto)
        COMMIT
        SELECT Puesto,IdPuestoEmpleado
        FROM PuestoEmpleado
        END
ELSE
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'El puesto ya existe'
        SELECT @error as Error
        END
END
GO

create procedure [dbo].[insertarSucursal]
    @Nombre varchar(50),
    @Descripcion varchar(50),
    @IdUbicacion int
AS
BEGIN
    declare @SucursalRepetida varchar(50)
    declare @UbicacionConfirmada int
    declare @UbicacionRepetida int

    SELECT @SucursalRepetida = Nombre
    FROM Sucursal
    WHERE Nombre = @Nombre

    SELECT @UbicacionRepetida = IdUbicacion
    FROM Sucursal
    WHere IdUbicacion = @IdUbicacion

    SELECT @UbicacionConfirmada = IdUbicacion
    FROM Ubicacion
    WHERE IdUbicacion = @IdUbicacion

    IF @SucursalRepetida IS NULL AND @UbicacionConfirmada IS NOT NULL AND @UbicacionRepetida is
null
        BEGIN
            BEGIN TRANSACTION
            insert into Sucursal(Nombre,Descripcion,IdUbicacion)
            values (@Nombre,@Descripcion,@UbicacionConfirmada)
            COMMIT
            SELECT Nombre,Descripcion,IdUbicacion
            FROM Sucursal
            END
        ELSE
            BEGIN
                DECLARE @error varchar(50)
                SET @error = 'La sucursal ya existe en la base o esa ubicacion esa ocupada'

```



```

                SELECT @error as Error
            END
END
GO

CREATE PROCEDURE [dbo].[InsertarTipoPago](@Pago VARCHAR(50))
AS
    IF @Pago IS NULL
        RAISERROR(N'Informacion Incompleta', 1, 1);
    ELSE
        BEGIN
            BEGIN TRANSACTION
            INSERT INTO TipoPago(Pago)
            VALUES(@Pago)
            COMMIT TRANSACTION
            RETURN @@IDENTITY;
        END
GO

create procedure [dbo].[insertarTipoVehiculo]
    @Nombre varchar(50)
AS
BEGIN
    declare @TipoRepetido varchar(50)

    SELECT @TipoRepetido = Nombre
    FROM TipoVehiculo
    WHERE Nombre = @Nombre

    IF @TipoRepetido IS NULL
        BEGIN
            BEGIN TRANSACTION
            insert into TipoVehiculo(Nombre)
            values (@Nombre)
            COMMIT
            SELECT Nombre,IdTipoVehiculo
            FROM TipoVehiculo
        END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El tipo ya existe'
            SELECT @error as Error
        END
END
GO

create procedure [dbo].[insertarUbicacion]
    @Descripcion varchar(50),
    @Locacion geometry,

```

```

                                @NombreProvincia varchar(50)
AS
BEGIN
    declare @UbicacionRepetida varchar(50)
    declare @ProvinciaConfirmado int

    SELECT @UbicacionRepetida = Descripcion
    FROM Ubicacion
    WHERE Descripcion = @Descripcion

    SELECT @ProvinciaConfirmado = IdProvincia
    FROM Provincia
    WHERE Nombre = @NombreProvincia

    IF @UbicacionRepetida IS NULL AND @ProvinciaConfirmado IS NOT NULL
    BEGIN
        BEGIN TRANSACTION
        insert into Ubicacion(Descripcion,IdProvincia,LocacionExacta)
        values (@Descripcion,@ProvinciaConfirmado,@Locacion)
        COMMIT
        SELECT @@IDENTITY AS IdUbicacion
    END
    ELSE
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'La ubicacion ya existe en la base'
        SELECT @error as Error
    END
END
GO

create procedure [dbo].[insertarVehiculo]
    @Tipo varchar(50),
                                @Combustible varchar(50),
                                @Marca varchar(50),
                                @Modelo varchar(50),
                                @Precio int,
                                @usado bit,
                                @puertas int
AS
BEGIN
    declare @TipoConfirmado int
    declare @CombustibleConfirmado int

    SELECT @TipoConfirmado = IdTipoVehiculo
    FROM TipoVehiculo
    WHERE Nombre = @Tipo

    SELECT @CombustibleConfirmado = IdCombustible
    FROM Combustible

```

```

WHERE Nombre = @Combustible

IF @TipoConfirmado IS NOT NULL AND @CombustibleConfirmado IS NOT NULL
    BEGIN
        BEGIN TRANSACTION
        insert into Vehiculo(IdTipoVehiculo,IdCombustible,Marca,Modelo,Precio, Usado,Puertas)
        values (@TipoConfirmado,@CombustibleConfirmado,@Marca,@Modelo,@Precio,
@usado,@puertas)
        COMMIT
        SELECT IdVehiculo,IdTipoVehiculo,IdCombustible,Marca,Modelo,Precio
        FROM Vehiculo
        END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El combustible o tipo indicado no existen'
            SELECT @error as Error
            END
END
GO

CREATE PROCEDURE [dbo].[InsertarVehiculoFabrica]
    @IdVehiculo int, @CostoVehiculo int, @IdInventario int, @IdFabrica int
AS
BEGIN
    Begin Transaction;
    --validar si existe
    insert into VehiculoFabrica(IdVehiculo,CostoVehiculo,IdInventario,IdFabrica)
    values(@IdVehiculo,@CostoVehiculo,@IdInventario,@IdFabrica)
    --end
    commit
END
GO

CREATE PROCEDURE [dbo].[InsertarVehiculoFabricaXEmpleado]
    @IdVehiculoFabrica int, @IdEmpleado int, @Labor Varchar(50), @FechaInicio Date, @FechaFinal
Date,@Estatus Varchar(50)
AS
BEGIN
    Begin Transaction;
    --validar si existe
    insert into
VehiculoFabricaXEmpleado(IdVehiculoFabrica,IdEmpleado,Labor,FechaInicio,FechaFinal,Estatus)
    values(@IdVehiculoFabrica,@IdEmpleado,@Labor,@FechaInicio,@FechaFinal,@Estatus)
    --end
    commit
END
GO

create procedure [dbo].[insertarVehiculoXSucursal]

```

```

                                @idVehiculo int,
                                @idSucursal int
AS
BEGIN
    declare @VehiculoConfirmado int
    declare @SucursalConfirmada int

    SELECT @VehiculoConfirmado = IdVehiculo
    FROM Vehiculo
    WHERE IdVehiculo = @idVehiculo

    SELECT @SucursalConfirmada = IdSucursal
    FROM Sucursal
    WHERE IdSucursal = @idSucursal

    IF @VehiculoConfirmado IS NOT NULL AND @SucursalConfirmada IS NOT NULL
        BEGIN
            BEGIN TRANSACTION
            insert into VehiculoXSucursal(IdSucursal,IdVehiculo)
            Values (@SucursalConfirmada,@VehiculoConfirmado)
            COMMIT
            SELECT IdSucursal,IdVehiculo
            FROM VehiculoXSucursal
            END
        ELSE
            BEGIN
            DECLARE @error varchar(100)
            SET @error = 'El vehiculo o sucursal indicado no existen'
            SELECT @error as Error
            END
    END
GO

create procedure [dbo].[loginUsuario]
    @username varchar(50),
    @password varchar(50)
AS

    DECLARE @IdUsuario VARCHAR(100)
    DECLARE @Privilegio VARCHAR(100)

    SELECT @IdUsuario = Usuario.Usuario, @Privilegio = Usuario.Privilegio
    FROM Usuario
    WHERE Usuario.Usuario = @username AND Usuario.Pass = @password

    IF @IdUsuario IS NULL
        BEGIN
            Select 'no existe el usuario' as error
        END

```

```

IF @Privilegio = 'Cliente'
BEGIN
    SELECT TOP 1 Cliente.IdCliente as IdUsuario, 'Cliente' as Privilegio
    FROM Cliente
    INNER JOIN Usuario
    ON Usuario.IdUsuario = Cliente.IdUsuario
END

IF @Privilegio = 'Empleado'
BEGIN
    SELECT TOP 1 Empleado.IdEmpleado as IdUsuario, 'Empleado' as Privilegio
    FROM Empleado
    INNER JOIN Usuario
    ON Usuario.IdUsuario = Empleado.IdUsuario
END

IF @Privilegio = 'Administrador'
BEGIN
    SELECT TOP 1 Empleado.IdEmpleado as IdUsuario, 'Administrador' as Privilegio
    FROM Empleado
    INNER JOIN Usuario
    ON Usuario.IdUsuario = Empleado.IdUsuario
END

GO

create procedure [dbo].[modificarCaracteristica]
    @Dato varchar(50),
    @NuevoDato varchar(50),
    @Caracteristica varchar(50)
AS
BEGIN
    declare @DatoConfirmado int

    SELECT @DatoConfirmado = IdCaracteristica
    FROM Caracteristica
    WHERE Dato = @Dato

    IF @DatoConfirmado IS NULL
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'La caracteristica no existe'
        SELECT @error as Error
    END
    ELSE
    BEGIN
        BEGIN TRANSACTION
        UPDATE Caracteristica
        SET Dato = isnull(@NuevoDato,Dato), Caracteristica = isnull(@Caracteristica,Caracteristica)
    END

```

```

        WHERE IdCaracteristica = @DatoConfirmado
        COMMIT
        SELECT Caracteristica,Dato,IdCaracteristica
        FROM Caracteristica
        END
END
GO

create procedure [dbo].[modificarCaracteristicaXVehiculo]
        @idCaracteristicaXVehiculo int,
        @idCaracteristica int,
        @idVehiculo int
AS
BEGIN
        declare @VehiculoConfirmado int
        declare @CaracteristicaConfirmado int
        declare @CaracteristicaXVehiculoConfirmado int

        SELECT @VehiculoConfirmado = IdVehiculo
        FROM Vehiculo
        WHERE IdVehiculo = @idVehiculo

        SELECT @CaracteristicaConfirmado = IdCaracteristica
        FROM Caracteristica
        WHERE IdCaracteristica = @idCaracteristica

        SELECT @CaracteristicaXVehiculoConfirmado = IdCaracteristicaXVehiculo
        FROM CaracteristicaXVehiculo
        WHERE IdCaracteristicaXVehiculo = @idCaracteristicaXVehiculo

        IF @CaracteristicaXVehiculoConfirmado IS NOT NULL
        BEGIN
                BEGIN TRANSACTION
                UPDATE CaracteristicaXVehiculo
                SET IdCaracteristica = isnull(@CaracteristicaConfirmado,IdCaracteristica),IdVehiculo =
isnull(@VehiculoConfirmado,IdVehiculo)
                WHERE IdCaracteristicaXVehiculo = @CaracteristicaXVehiculoConfirmado
                COMMIT
                SELECT IdCaracteristica,IdVehiculo
                FROM CaracteristicaXVehiculo
                END
        ELSE
        BEGIN
                DECLARE @error varchar(50)
                SET @error = 'El CaracteristicaXVehiculo no existe'
                SELECT @error as Error
                END
        END
GO

```

```

create procedure [dbo].[modificarCombustible]
    @Nombre varchar(50),
    @NuevoNombre varchar(50)
AS
BEGIN
    declare @CombustibleConfirmado int

    SELECT @CombustibleConfirmado = IdCombustible
    FROM Combustible
    WHERE Nombre = @Nombre

    IF @CombustibleConfirmado IS NULL
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'El combustible no existe'
        SELECT @error as Error
    END
    ELSE
    BEGIN
        BEGIN TRANSACTION
        UPDATE Combustible
        SET Nombre = isnull(@NuevoNombre,Nombre)
        WHERE IdCombustible = @CombustibleConfirmado
        COMMIT
        SELECT Nombre,IdCombustible
        FROM Combustible
    END
END
GO

```

```

create procedure [dbo].[modificarEmpleado]
    @Nombre varchar(50),
    @Apellido varchar(50),
    @Telefono varchar(50),
    @Correo varchar(50),
    @Supervisor varchar(50),
    @Puesto varchar(50),
    @Sucursal int,
    @Cedula varchar(50)

```

```

AS
BEGIN
    declare @CedulaRepetida varchar(50)
    declare @SucursalConfirmada int
    declare @SupervisorConfirmado int
    declare @PuestoConfirmado int

    SELECT @CedulaRepetida = Cedula
    FROM Empleado
    WHERE Cedula = @Cedula

```

```

SELECT @SucursalConfirmada = IdSucursal
FROM Sucursal
WHERE IdSucursal = @Sucursal

SELECT @SupervisorConfirmado = IdEmpleado
FROM Empleado
WHERE Cedula = @Supervisor

SELECT @PuestoConfirmado = IdPuestoEmpleado
FROM PuestoEmpleado
WHERE Puesto = @Puesto

IF @CedulaRepetida IS NOT NULL
    BEGIN
        BEGIN TRANSACTION
        UPDATE Empleado
        SET Nombre = isnull(@Nombre,Nombre),Apellido = isnull(@Apellido,Apellido),Telefono =
isnull(@Telefono,Telefono),Correo = isnull(@Correo,Correo),
        IdSupervisor = isnull(@SupervisorConfirmado,IdSupervisor),IdPuestoEmpleado =
isnull(@PuestoConfirmado,IdPuestoEmpleado)
        Where Cedula = @CedulaRepetida
        COMMIT
        SELECT IdEmpleado,Nombre,Apellido,Telefono,Correo,IdSupervisor,IdPuestoEmpleado,Cedula
        FROM Empleado
        END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'La cedula no existe'
            SELECT @error as Error
            END
END
GO

create procedure [dbo].[modificarExtra]
    @Nombre varchar(50),
    @NuevoNombre varchar(50),
    @Descripcion varchar(50),
    @Precio int
AS
BEGIN
    declare @ExtraConfirmado int

    SELECT @ExtraConfirmado = IdExtra
    FROM Extra
    WHERE Nombre = @Nombre

    IF @ExtraConfirmado IS NULL
        BEGIN

```



```

        DECLARE @error varchar(50)
        SET      @error = 'El extra no existe'
        SELECT @error as Error
        END
ELSE
    BEGIN
        BEGIN TRANSACTION
        UPDATE Extra
        SET Nombre = isnull(@NuevoNombre,Nombre),Descripcion = isnull(@Descripcion,Descripcion),
Precio = isnull(@Precio,Precio)
        WHERE IdExtra = @ExtraConfirmado
        COMMIT
        SELECT Nombre,Descripcion,Precio,IdExtra
        FROM Extra
        END
END
GO

create procedure [dbo].[modificarExtraXVehiculo]
        @idExtraXVehiculo int,
        @idExtra int,
        @idVehiculo int,
        @precio int
AS
BEGIN
    declare @VehiculoConfirmado int
    declare @ExtraConfirmado int
    declare @ExtraXVehiculoConfirmado int

    SELECT @VehiculoConfirmado = IdVehiculo
    FROM Vehiculo
    WHERE IdVehiculo = @idVehiculo

    SELECT @ExtraConfirmado = IdExtra
    FROM Extra
    WHERE IdExtra = @idExtra

    SELECT @ExtraXVehiculoConfirmado = IdExtraXVehiculo
    FROM ExtraXVehiculo
    WHERE IdExtraXVehiculo = @idExtraXVehiculo

    IF @ExtraXVehiculoConfirmado IS NOT NULL
        BEGIN
            BEGIN TRANSACTION
            UPDATE ExtraXVehiculo
            SET PrecioPagado = isnull(@precio,PrecioPagado),IdExtra =
isnull(@ExtraConfirmado,IdExtra),IdVehiculo = isnull(@VehiculoConfirmado,IdVehiculo)
            WHERE IdExtraXVehiculo = @ExtraXVehiculoConfirmado
            COMMIT
            SELECT IdExtra,IdVehiculo,PrecioPagado

```

```

        FROM ExtraXVehiculo
    END
ELSE
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'El extraXVehiculo no existe'
        SELECT @error as Error
    END
END
GO

CREATE PROCEDURE [dbo].[ModificarFabrica]
    @IdFabrica int, @Descripcion Varchar(50),@IdUbicacion int
AS
BEGIN
    BEGIN TRANSACTION
        update Fabrica
        SET Descripcion = isnull(@Descripcion,Descripcion),IdUbicacion =
isnull(@IdUbicacion,IdUbicacion)
        WHERE IdFabrica = @IdFabrica
    COMMIT
END
GO

create procedure [dbo].[modificarFotoVehiculo]
    @IdFotoVehiculo int,
    @foto varchar(50),
    @Nombre varchar(50),
    @fecha date,
    @idVehiculo int
AS
BEGIN
    declare @VehiculoConfirmado int
    declare @fotoConfirmada int
    declare @sql varchar(max)

    SELECT @fotoConfirmada = IdFotoVehiculo
    FROM FotoVehiculo
    WHERE IdFotoVehiculo = @IdFotoVehiculo

    SELECT @VehiculoConfirmado = IdVehiculo
    FROM Vehiculo
    WHERE IdVehiculo = @idVehiculo

    set @sql = 'UPDATE FotoVehiculo SET Foto = (SELECT BulkColumn FROM OPENROWSET(BULK
''+@foto+'', SINGLE_BLOB) as X) WHERE IdFotoVehiculo =' + Cast(@fotoConfirmada as varchar(50))

    IF @fotoConfirmada is null
    BEGIN
        DECLARE @error varchar(50)

```

```

        SET      @error = 'El vehiculo indicado no existe'
        SELECT @error as Error
        END

    ELSE IF @foto is not null
    BEGIN
        BEGIN TRANSACTION
        exec (@sql)
        UPDATE FotoVehiculo
        SET NombreFoto = isnull(@Nombre,NombreFoto),Fecha = isnull(@fecha,Fecha),IdVehiculo =
isnull(@VehiculoConfirmado,IdVehiculo)
        WHERE IdFotoVehiculo = @fotoConfirmada
        COMMIT
        SELECT IdFotoVehiculo,Foto,NombreFoto,Fecha,IdVehiculo
        FROM FotoVehiculo
        END
    ELSE IF @foto is null
    BEGIN
        BEGIN TRANSACTION
        UPDATE FotoVehiculo
        SET NombreFoto = isnull(@Nombre,NombreFoto),Fecha = isnull(@fecha,Fecha),IdVehiculo =
isnull(@VehiculoConfirmado,IdVehiculo)
        WHERE IdFotoVehiculo = @fotoConfirmada
        COMMIT
        SELECT IdFotoVehiculo,Foto,NombreFoto,Fecha,IdVehiculo
        FROM FotoVehiculo
        END
    END
GO

CREATE  PROCEDURE [dbo].[ModificarInventario]
    @IdInventario int, @NombreInventario varchar(50), @Fecha date, @IdSucursal int
AS
BEGIN
    BEGIN TRANSACTION
        update Inventario
        SET NombreInventario=ISNULL(@NombreInventario,NombreInventario),
            Fecha=ISNULL(@Fecha,Fecha),
            IdSucursal = isnull(@IdSucursal,IdSucursal)
        WHERE IdInventario = @IdInventario
    COMMIT
END
GO

create  procedure [dbo].[modificarPais]
    @Nombre varchar(50),
    @NuevoNombre varchar(50),
    @Locacion geometry
AS
BEGIN

```

```

declare @PaisRepetido int
SELECT @PaisRepetido = IdPais
FROM Pais
WHERE Nombre = @Nombre

IF @PaisRepetido IS NULL
BEGIN
    DECLARE @error varchar(50)
    SET @error = 'El pais no existe en la base'
    SELECT @error as Error
END
ELSE
BEGIN

    BEGIN TRANSACTION
    UPDATE Pais
    SET Nombre = isnull(@NuevoNombre,Nombre),Locacion = isnull(@Locacion,Locacion)
    WHERE IdPais = @PaisRepetido
    COMMIT
    SELECT Nombre,Locacion,IdPais
    FROM Pais
    END

END
GO

CREATE PROCEDURE [dbo].[ModificarPedido]
    @IdPedido int, @EstatusPedido Varchar(50), @FechaPedido Date, @IdEmpleadoPedido int, @IdFabrica
int
AS
BEGIN
    BEGIN TRANSACTION
    update Pedido
    SET EstatusPedido = isnull(@EstatusPedido,EstatusPedido),
        FechaPedido = isnull(@FechaPedido,FechaPedido),
        IdEmpleadoPedido=ISNULL(@IdEmpleadoPedido,IdEmpleadoPedido),
        IdFabrica=ISNULL(@IdFabrica,IdFabrica)
    WHERE IdPedido = @IdPedido
    COMMIT
END
GO

CREATE PROCEDURE [dbo].[ModificarPedidoXVehiculo]
    @IdPedidoXVehiculo int, @IdPedido int, @IdVehiculo int, @FechaCompletado date
AS
BEGIN
    BEGIN TRANSACTION
    update PedidoXVehiculo
    SET IdPedido = isnull(@IdPedido,IdPedido),
        IdVehiculo = isnull(@IdVehiculo,IdVehiculo),

```

```

                FechaCompletado=ISNULL(@FechaCompletado,FechaCompletado)
            WHERE IdPedidoXVehiculo = @IdPedidoXVehiculo
        COMMIT
    END
    GO

create procedure [dbo].[modificarProvincia]
    @Nombre varchar(50),
    @NuevoNombre varchar(50),
    @Locacion geometry,
    @idPais int
AS
BEGIN
    declare @ProvinciaRepetida int
    declare @PaisConfirmado int

    SELECT @ProvinciaRepetida = IdProvincia
    FROM Provincia
    WHERE Nombre = @Nombre

    SELECT @PaisConfirmado = IdPais
    FROM Pais
    WHERE IdPais = @idPais

    IF @ProvinciaRepetida IS NULL
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El pais o la provincia no existe en la base'
            SELECT @error as Error
        END
    ELSE
        BEGIN

            BEGIN TRANSACTION
            UPDATE Provincia
            SET Nombre = isnull(@NuevoNombre,Nombre),LocacionProvincia =
isnull(@Locacion,LocacionProvincia),IdPais = isnull(@PaisConfirmado,IdPais)
            WHERE IdProvincia = @ProvinciaRepetida
            COMMIT
            SELECT Nombre,LocacionProvincia,IdPais,IdProvincia
            FROM Provincia
        END
    END
    GO

create procedure [dbo].[modificarPuestoEmpleado]
    @Puesto varchar(50),
    @NuevoPuesto varchar(50)

```

```

AS
BEGIN
    declare @PuestoRepetido int

    SELECT @PuestoRepetido = IdPuestoEmpleado
    FROM PuestoEmpleado
    WHERE Puesto = @Puesto

    IF @PuestoRepetido IS NOT NULL
        BEGIN
            BEGIN TRANSACTION
            UPDATE PuestoEmpleado
            SET Puesto = isnull(@NuevoPuesto,Puesto)
            WHERE IdPuestoEmpleado = @PuestoRepetido
            COMMIT
            SELECT Puesto,IdPuestoEmpleado
            FROM PuestoEmpleado
        END
    ELSE
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El puesto no existe'
            SELECT @error as Error
        END
END
GO

create procedure [dbo].[modificarSucursal]
    @Nombre varchar(50),
    @NuevoNombre varchar(50),
    @Descripcion varchar(50),
    @IdUbicacion int
AS
BEGIN
    declare @SucursalRepetida varchar(50)
    declare @UbicacionConfirmada int
    declare @UbicacionRepetida int

    SELECT @SucursalRepetida = Nombre
    FROM Sucursal
    WHERE Nombre = @Nombre

    SELECT @UbicacionRepetida = IdUbicacion
    FROM Sucursal
    WHERE IdUbicacion = @IdUbicacion

    SELECT @UbicacionConfirmada = IdUbicacion
    FROM Ubicacion
    WHERE IdUbicacion = @IdUbicacion

```

```

        IF @SucursalRepetida IS NOT NULL AND @UbicacionConfirmada IS NOT NULL AND
        @UbicacionRepetida is null
            BEGIN
                BEGIN TRANSACTION
                UPDATE Sucursal
                SET Nombre = isnull(@Nombre,Nombre),Descripcion =
isnull(@Descripcion,Descripcion),IdUbicacion = isnull(@IdUbicacion,IdUbicacion)
                COMMIT
                SELECT Nombre,Descripcion,IdUbicacion
                FROM Sucursal
            END
        ELSE
            BEGIN
                DECLARE @error varchar(50)
                SET @error = 'La sucursal no existe en la base o esa ubicacion esa ocupada'
                SELECT @error as Error
            END
END
GO

create procedure [dbo].[modificarTipoVehiculo]
    @Nombre varchar(50),
    @NuevoNombre varchar(50)
AS
BEGIN
    declare @TipoConfirmado int

    SELECT @TipoConfirmado = IdTipoVehiculo
    FROM TipoVehiculo
    WHERE Nombre = @Nombre

    IF @TipoConfirmado IS NULL
        BEGIN
            DECLARE @error varchar(50)
            SET @error = 'El tipo no existe'
            SELECT @error as Error
        END
    ELSE
        BEGIN
            BEGIN TRANSACTION
            UPDATE TipoVehiculo
            SET Nombre = isnull(@NuevoNombre,Nombre)
            WHERE IdTipoVehiculo = @TipoConfirmado
            COMMIT
            SELECT Nombre,IdTipoVehiculo
            FROM TipoVehiculo
        END
END
GO

```

```

create procedure [dbo].[modificarUbicacion]
    @Descripcion varchar(50),
    @NuevaDescripcion varchar(50),
    @Locacion geometry,
    @idProvincia int
AS
BEGIN
    declare @UbicacionRepetida int
    declare @ProvinciaConfirmada int

    SELECT @UbicacionRepetida = IdUbicacion
    FROM Ubicacion
    WHERE Descripcion = @Descripcion

    SELECT @ProvinciaConfirmada = IdProvincia
    FROM Provincia
    WHERE IdProvincia = @idProvincia

    IF @UbicacionRepetida IS NULL
    BEGIN
        DECLARE @error varchar(50)
        SET @error = 'La ubicacion o la provincia no existe en la base'
        SELECT @error as Error
    END
    ELSE
    BEGIN
        BEGIN TRANSACTION
        UPDATE Ubicacion
        SET Descripcion = isnull(@NuevaDescripcion, Descripcion), LocacionExacta =
isnull(@Locacion, LocacionExacta), IdProvincia = isnull(@ProvinciaConfirmada, IdProvincia)
        WHERE IdUbicacion = @UbicacionRepetida
        COMMIT
        SELECT Descripcion, LocacionExacta, IdUbicacion, IdProvincia
        FROM Ubicacion
    END
END
GO

create procedure [dbo].[modificarVehiculo]
    @idVehiculo int,
    @Tipo varchar(50),
    @Combustible varchar(50),
    @Marca varchar(50),
    @Modelo varchar(50),
    @Precio int,
    @Usado bit,
    @Puertas int
AS
BEGIN

```



```

declare @VehiculoConfirmado int
declare @TipoConfirmado int
declare @CombustibleConfirmado int

SELECT @TipoConfirmado = IdTipoVehiculo
FROM TipoVehiculo
WHERE Nombre = @Tipo

SELECT @CombustibleConfirmado = IdCombustible
FROM Combustible
WHERE Nombre = @Combustible

SELECT @VehiculoConfirmado = IdVehiculo
FROM Vehiculo
WHERE IdVehiculo = @IdVehiculo

IF @VehiculoConfirmado IS NULL
BEGIN
    DECLARE @error varchar(50)
    SET @error = 'El vehiculo tipo o combustible no existe'
    SELECT @error as Error
END
ELSE
BEGIN
    BEGIN TRANSACTION
    UPDATE Vehiculo
    SET IdTipoVehiculo = isnull(@TipoConfirmado,IdTipoVehiculo),IdCombustible =
isnull(@CombustibleConfirmado,IdCombustible),Marca = isnull(@Marca,Marca),Modelo = isnull(@Modelo,Modelo),
        Precio = isnull(@Precio,Precio),Usado = ISNULL(@Usado,Usado),Puertas =
ISNULL(@Puertas,Puertas)
    WHERE IdVehiculo = @VehiculoConfirmado
    COMMIT
    SELECT IdVehiculo,IdTipoVehiculo,IdCombustible,Marca,Modelo,Precio
    FROM Vehiculo
END

END
GO

CREATE PROCEDURE [dbo].[ModificarVehiculoFabrica]
    @IdVehiculoFabrica int, @IdVehiculo int, @CostoVehiculo int, @IdInventario int, @IdFabrica int
AS
BEGIN
    BEGIN TRANSACTION
        update VehiculoFabrica
        SET IdVehiculo=ISNULL(@IdVehiculo,IdVehiculo),
            CostoVehiculo=ISNULL(@CostoVehiculo,CostoVehiculo),
            IdInventario=ISNULL(@IdInventario,IdInventario),
            IdFabrica = isnull(@IdFabrica,IdFabrica)
        WHERE IdVehiculoFabrica = @IdVehiculoFabrica
    COMMIT

```

```

END
GO

CREATE PROCEDURE [dbo].[ModificarVehiculoFabricaXEmpleado]
    @IdVehiculoFabricaXEmpleado int,@IdVehiculoFabrica int, @IdEmpleado int, @Labor Varchar(50),
    @FechaInicio Date, @FechaFinal Date,@Estatus Varchar(50)
AS
BEGIN
    BEGIN TRANSACTION
        update VehiculoFabricaXEmpleado
        SET IdVehiculoFabrica=ISNULL(@IdVehiculoFabrica,IdVehiculoFabrica),
            IdEmpleado=ISNULL(@IdEmpleado,IdEmpleado),
            Labor=Isnull(@Labor,Labor),
            FechaInicio=ISNULL(@FechaInicio,FechaInicio),
            FechaFinal=ISNULL(@FechaFinal,FechaFinal),
            Estatus=ISNULL(@Estatus,Estatus)
        WHERE IdVehiculoFabricaXEmpleado = @IdVehiculoFabricaXEmpleado
    COMMIT
END
GO

create procedure [dbo].[modificarVehiculoXSucursal]
    @idVehiculoXSucursal int,
    @idSucursal int,
    @idVehiculo int
AS
BEGIN
    declare @VehiculoConfirmado int
    declare @SucursalConfirmada int
    declare @VehiculoXSucursalConfirmado int

    SELECT @VehiculoConfirmado = IdVehiculo
    FROM Vehiculo
    WHERE IdVehiculo = @idVehiculo

    SELECT @SucursalConfirmada = IdSucursal
    FROM Sucursal
    WHERE IdSucursal = @idSucursal

    SELECT @VehiculoXSucursalConfirmado = IdVehiculoXSucursal
    FROM VehiculoXSucursal
    WHERE IdVehiculoXSucursal = @idVehiculoXSucursal

    IF @VehiculoXSucursalConfirmado IS NOT NULL
    BEGIN
        BEGIN TRANSACTION
        UPDATE VehiculoXSucursal
        SET IdSucursal = isnull(@SucursalConfirmada,IdSucursal),IdVehiculo =
isnull(@VehiculoConfirmado,IdVehiculo)
        WHERE IdVehiculoXSucursal = @VehiculoXSucursalConfirmado
    
```

```

        COMMIT
        SELECT IdExtra,IdVehiculo,PrecioPagado
        FROM ExtraXVehiculo
        END
ELSE
    BEGIN
        DECLARE @error varchar(50)
        SET      @error = 'El extraXVehiculo no existe'
        SELECT @error as Error
        END
END
GO

CREATE  PROCEDURE [dbo].[ObtenerDetalleFactura](@IdFactura INT)
AS
    SELECT SubTotal, Comentario, Cliente.Nombre AS NombreCliente, Empleado.Nombre AS
EmpleadoNombre, TipoPago.Pago as TipoPago, Sucursal.Nombre as Sucursal
    FROM DetalleFactura
    INNER JOIN Cliente
    ON Cliente.IdCliente = DetalleFactura.IdCliente
    INNER JOIN Empleado
    ON Empleado.IdEmpleado = DetalleFactura.IdVendedor
    INNER JOIN Factura
    ON Factura.IdDetalleFactura = DetalleFactura.IdDetalleFactura
    INNER JOIN TipoPago
    ON DetalleFactura.IdTipoPago = TipoPago.IdTipoPago
    INNER JOIN Sucursal
    ON DetalleFactura.IdSucursal = Sucursal.IdSucursal
    WHERE IdFactura = @IdFactura
GO

create  procedure [dbo].[seleccionarCaracteristica]
        @Caracteristica varchar(50),
        @Dato varchar(50)
AS
BEGIN

    SELECT IdCaracteristica,Caracteristica,Dato
    FROM Caracteristica
    WHERE Caracteristica.Caracteristica = isnull(@Caracteristica,Caracteristica.Caracteristica) AND
Caracteristica.Dato = isnull(@Dato,Caracteristica.Dato)
END
GO

create  procedure [dbo].[seleccionarCaracteristicaXVehiculo]
        @idCaracteristica int,
        @idVehiculo int
AS
BEGIN

```

```

SELECT
Vehiculo.Marca,Vehiculo.Modelo,Caracteristica.Dato,CaracteristicaXVehiculo.IdCaracteristicaXVehiculo
FROM  CaracteristicaXVehiculo inner join Vehiculo on CaracteristicaXVehiculo.IdVehiculo =
Vehiculo.IdVehiculo
inner join Caracteristica on CaracteristicaXVehiculo.IdCaracteristica = Caracteristica.IdCaracteristica
WHERE CaracteristicaXVehiculo.IdCaracteristica =
isnull(@IdCaracteristica,CaracteristicaXVehiculo.IdCaracteristica) AND
CaracteristicaXVehiculo.IdVehiculo = isnull(@IdVehiculo,CaracteristicaXVehiculo.IdVehiculo)

END
GO

```

```

CREATE PROCEDURE [dbo].[SeleccionarClientes](@Nombre VARCHAR(50), @Apellido VARCHAR(50),
@Telefono VARCHAR(50), @Correo VARCHAR(50), @Provincia VARCHAR(50), @Pais VARCHAR(50))
AS

```

```

SELECT Cliente.Nombre, Apellido, Telefono, Correo,Ubicacion.Descripcion
FROM Cliente
INNER JOIN Ubicacion
ON Ubicacion.IdUbicacion = Cliente.IdUbicacion
INNER JOIN Provincia
ON Provincia.IdProvincia = Ubicacion.IdProvincia
INNER JOIN Pais
ON Provincia.IdPais = Pais.IdPais
WHERE Cliente.Nombre = ISNULL(@Nombre, Cliente.Nombre) AND
Apellido = ISNULL(@Apellido, Apellido) AND
Telefono = ISNULL(@Telefono, Telefono) AND
Correo = ISNULL(@Correo, Correo) AND
Provincia.Nombre = ISNULL(@Provincia, Provincia.Nombre) AND
Pais.Nombre = ISNULL(@Pais, Pais.Nombre);

```

GO

```

create procedure [dbo].[seleccionarCombustible]
@Nombre varchar(50)

```

AS

BEGIN

```

SELECT Nombre,IdCombustible
FROM Combustible
WHERE Nombre = isnull(@Nombre,Nombre)

```

END

GO

```

CREATE PROCEDURE [dbo].[SeleccionarComisiones](@IdComision INT, @ComisionInicial INT, @ComisionFinal
INT, @FechaInicial DATE, @FechaFinal DATE)

```

AS

```

SELECT Comision, Empleado.Nombre, Factura.Fecha
FROM Comision
INNER JOIN Empleado
ON Comision.IdEmpleado = Empleado.IdEmpleado
INNER JOIN Factura

```

```

        ON Comision.IdFactura = Factura.IdFactura
        WHERE IdComision = ISNULL(@IdComision, IdComision) AND
              Comision.Comision BETWEEN ISNULL(@ComisionInicial, Comision.Comision) AND
ISNULL(@ComisionFinal, Comision.Comision) AND
              Factura.Fecha BETWEEN ISNULL(@FechaInicial, Fecha) AND ISNULL(@FechaFinal,
Fecha)
GO

```

```

CREATE PROCEDURE [dbo].[SeleccionarDescuentos](@IdFactura INT)
AS

```

```

        SELECT Nombre, Descuento.Descuento
        FROM Descuento
        INNER JOIN DescuentoXDetalleFactura
        ON Descuento.IdDescuento = DescuentoXDetalleFactura.IdDescuento
        INNER JOIN DetalleFactura
        ON DetalleFactura.IdDetalleFactura = DescuentoXDetalleFactura.IdDetalleFactura
        INNER JOIN Factura
        ON Factura.IdDetalleFactura = DetalleFactura.IdDetalleFactura
        WHERE IdFactura = @IdFactura
GO

```

```

CREATE PROCEDURE [dbo].[SeleccionarDescuentoXDetalleFactura](@IdDescuento INT, @IdDetalleFactura INT)
AS

```

```

        SELECT DescuentoXDetalleFactura.IdDescuento,
DescuentoXDetalleFactura.IdDetalleFactura, DetalleFactura.Comentario,
DetalleFactura.SubTotal, Descuento.Descuento as Descuento
        FROM DescuentoXDetalleFactura inner join Descuento on DescuentoXDetalleFactura.IdDescuento =
Descuento.IdDescuento
        inner join DetalleFactura on DescuentoXDetalleFactura.IdDetalleFactura =
DetalleFactura.IdDetalleFactura
        WHERE DescuentoXDetalleFactura.IdDescuento = ISNULL(@IdDescuento,
DescuentoXDetalleFactura.IdDescuento) AND
              DescuentoXDetalleFactura.IdDetalleFactura = ISNULL(@IdDetalleFactura,
DescuentoXDetalleFactura.IdDetalleFactura)
GO

```

```

create procedure [dbo].[seleccionarEmpleado]
        @Nombre varchar(50),
        @Apellido varchar(50),
        @Telefono varchar(50),
        @Correo varchar(50),
        @Supervisor varchar(50),
        @Puesto varchar(50),
        @Sucursal int,
        @Cedula varchar(50)

```

```

AS
BEGIN

```

```

SELECT
Empleado.IdEmpleado, Empleado.Nombre, Empleado.Apellido, Empleado.Telefono, Empleado.Correo, Empleado.IdSupervisor,
PuestoEmpleado.Puesto, Sucursal.Nombre, Empleado.Cedula

FROM Empleado INNER JOIN PuestoEmpleado on Empleado.IdPuestoEmpleado =
PuestoEmpleado.IdPuestoEmpleado
    inner join EmpleadoXSucursal on Empleado.IdEmpleado = EmpleadoXSucursal.IdSucursal
    inner join Empleado e2 on Empleado.IdSupervisor = e2.IdEmpleado
    , Sucursal

WHERE Sucursal.IdSucursal = EmpleadoXSucursal.IdSucursal AND
    Empleado.Nombre = isnull(@Nombre, Empleado.Nombre) AND Empleado.Apellido =
isnull(@Apellido, Empleado.Apellido) AND Empleado.Telefono = isnull(@Telefono, Empleado.Telefono)
    AND Empleado.Correo = isnull(@Correo, Empleado.Correo) AND e2.Cedula =
isnull(@Supervisor, e2.Cedula) AND PuestoEmpleado.Puesto = isnull(@Puesto, PuestoEmpleado.Puesto)
    AND Sucursal.IdSucursal = isnull(@Sucursal, Sucursal.IdSucursal) AND Empleado.Cedula =
isnull(@Cedula, Empleado.Cedula)

END
GO

CREATE PROCEDURE [dbo].[SeleccionarEmpleadosFabrica]
    @Nombre varchar(50),
    @Apellido varchar(50),
    @Telefono varchar(50),
    @Correo varchar(50),
    @Supervisor varchar(50),
    @Puesto varchar(50),
    @Fabrica int,
    @Cedula varchar(50)
AS
SELECT Empleado.Nombre, Apellido, Telefono, Correo, Fabrica.Descripcion
FROM Empleado, EmpleadoXFabrica, Fabrica
WHERE Empleado.IdEmpleado = EmpleadoXFabrica.IdEmpleado AND EmpleadoXFabrica.IdFabrica =
Fabrica.IdFabrica AND
    Empleado.Nombre = ISNULL(@Nombre, Empleado.Nombre) AND
    Apellido = ISNULL(@Apellido, Apellido) AND
    Telefono = ISNULL(@Telefono, Correo) AND
    Correo = ISNULL(@Correo, Correo) AND
    IdSupervisor = ISNULL(@Supervisor, IdSupervisor) AND
    IdPuestoEmpleado = ISNULL(@Puesto, IdPuestoEmpleado) AND
    Fabrica.IdFabrica = ISNULL(@Fabrica, Fabrica.IdFabrica) AND
    Empleado.Cedula = ISNULL(@Cedula, Empleado.Cedula)
GO

CREATE PROCEDURE [dbo].[SeleccionarEntregas](@IdEntrega INT, @EstatusEntrega VARCHAR(50),
@FechaInicial DATE, @FechaFinal DATE, @NumeroFactura VARCHAR(50))
AS
SELECT NumeroFactura, EstatusEntrega, Fecha, IdEntrega, Factura.PrecioTotal as
Monto, Ubicacion.Descripcion

```

```

FROM      Entrega
INNER JOIN Factura
ON          Factura.IdFactura = Entrega.IdFactura
INNER JOIN Ubicacion
ON      Entrega.IdUbicacionEntrega = Ubicacion.IdUbicacion
WHERE      IdEntrega = ISNULL(@IdEntrega, IdEntrega) AND
            EstatusEntrega = ISNULL(@EstatusEntrega, EstatusEntrega) AND
            Entrega.FechaEntrega BETWEEN ISNULL(@FechaInicial,
Entrega.FechaEntrega) AND ISNULL(@FechaFinal, Entrega.FechaEntrega) AND
            NumeroFactura = ISNULL(@NumeroFactura, NumeroFactura)

GO

create procedure [dbo].[seleccionarExtra]
    @Nombre varchar(50),
    @Descripcion varchar(50),
    @Precio int
AS
BEGIN
    SELECT Nombre,Descripcion,Precio,IdExtra
    FROM Extra
    WHERE Nombre = isnull(@Nombre,Nombre) AND Descripcion = isnull(@Descripcion,Descripcion) AND
Precio = isnull(@Precio,Precio)
END
GO

create procedure [dbo].[seleccionarExtraXVehiculo]
    @idExtra int,
    @idVehiculo int,
    @precioLow int,
    @precioHigh int
AS
BEGIN
    SELECT
Vehiculo.Marca,Vehiculo.Modelo,Extra.Nombre,ExtraXVehiculo.IdExtraXVehiculo,ExtraXVehiculo.PrecioPagado
FROM ExtraXVehiculo inner join Vehiculo on ExtraXVehiculo.IdVehiculo = Vehiculo.IdVehiculo
    inner join Extra on ExtraXVehiculo.IdExtra = Extra.IdExtra
    WHERE ExtraXVehiculo.IdExtra = isnull(@idExtra,ExtraXVehiculo.idExtra) AND
ExtraXVehiculo.IdVehiculo = isnull(@idVehiculo,ExtraXVehiculo.IdVehiculo) AND
    ExtraXVehiculo.PrecioPagado between isnull(@precioLow,ExtraXVehiculo.PrecioPagado) and
isnull(@precioHigh,ExtraXVehiculo.PrecioPagado)

END
GO

CREATE PROCEDURE [dbo].[SeleccionarFabrica]
    @IdFabrica int, @Descripcion Varchar(50),@NombreProvincia varchar(50),@NombrePais
varchar(50),@IdUbicacion int
AS
BEGIN

```

```

SELECT Fabrica.IdFabrica,Fabrica.Descripcion,Ubicacion.Descripcion,Provincia.Nombre,Pais.Nombre
FROM Fabrica inner join Ubicacion on Fabrica.IdUbicacion = Ubicacion.IdUbicacion,Provincia,Pais
WHERE      Ubicacion.IdProvincia = Provincia.IdProvincia AND Provincia.IdPais = Pais.IdPais AND
            IdFabrica=isnull(@IdFabrica,IdFabrica) AND Fabrica.Descripcion =
isnull(@Descripcion,Fabrica.Descripcion) AND
            Fabrica.IdUbicacion = isnull(@IdUbicacion,Fabrica.IdUbicacion) AND Provincia.Nombre =
ISNULL(@NombreProvincia,Provincia.Nombre) AND
            Pais.Nombre = ISNULL(@NombrePais,Pais.Nombre)
END
GO

CREATE  PROCEDURE [dbo].[SeleccionarFacturas](@IdFactura INT, @NumeroFactura VARCHAR(50),
@PrecioInicial INT, @PrecioFinal INT, @FechaInicial DATE, @FechaFinal DATE, @tipoPago VARCHAR(50))
AS
    SELECT NumeroFactura, PrecioTotal, Fecha, Factura.IdDetalleFactura,DetalleFactura.Comentario,
DetalleFactura.Descuento, TipoPago.Pago
    FROM  Factura inner join DetalleFactura on Factura.IdDetalleFactura = DetalleFactura.IdDetalleFactura
            inner join TipoPago on DetalleFactura.IdTipoPago =
TipoPago.IdTipoPago
    WHERE IdFactura = ISNULL(@IdFactura, IdFactura) AND
            NumeroFactura = ISNULL(@NumeroFactura, NumeroFactura) AND
            PrecioTotal BETWEEN ISNULL(@PrecioInicial, PrecioTotal) AND ISNULL(@PrecioFinal,
PrecioTotal) AND
            Fecha BETWEEN ISNULL(@FechaInicial, Fecha) AND ISNULL(@FechaFinal, Fecha)
AND
            TipoPago.Pago = ISNULL(@tipoPago, TipoPago.Pago)
GO

CREATE  PROCEDURE [dbo].[SeleccionarFacturaXVehiculo](@IdFactura INT, @IdVehiculo INT)
AS
    SELECT FacturaXVehiculo.IdFactura,
FacturaXVehiculo.IdVehiculo,Vehiculo.Marca,Vehiculo.Modelo,Factura.PrecioTotal
    FROM  FacturaXVehiculo Inner join Vehiculo on FacturaXVehiculo.IdVehiculo = Vehiculo.IdVehiculo
            inner join Factura on FacturaXVehiculo.IdFactura = Factura.IdFactura
    WHERE FacturaXVehiculo.IdFactura = ISNULL(@IdFactura, FacturaXVehiculo.IdFactura) AND
            FacturaXVehiculo.IdVehiculo = ISNULL(@IdVehiculo, FacturaXVehiculo.IdVehiculo)
GO

create  procedure [dbo].[seleccionarFotoVehiculo]
            @Nombre varchar(50),
            @fechaInicio date,
            @fechaFin date,
            @idVehiculo int
AS
BEGIN

    SELECT
IdFotoVehiculo,Foto,NombreFoto,Fecha,FotoVehiculo.IdVehiculo,Vehiculo.Modelo,Vehiculo.Marca
    FROM FotoVehiculo inner join Vehiculo on FotoVehiculo.IdVehiculo = Vehiculo.IdVehiculo
    WHERE FotoVehiculo.NombreFoto = isnull(@Nombre,FotoVehiculo.NombreFoto) AND

```



```

FotoVehiculo.Fecha between isnull(@fechaInicio,FotoVehiculo.Fecha) AND
isnull(@fechaFin,FotoVehiculo.Fecha) AND
FotoVehiculo.IdVehiculo = isnull(@idVehiculo,FotoVehiculo.IdVehiculo)

```

```

END
GO

```

```

CREATE PROCEDURE [dbo].[SeleccionarImpuestos](@IdFactura INT)
AS
    SELECT Nombre, Impuesto
    FROM Impuesto
    INNER JOIN ImpuestoXDetalleFactura
    ON ImpuestoXDetalleFactura.IdImpuesto = Impuesto.IdImpuesto
    INNER JOIN DetalleFactura
    ON DetalleFactura.IdDetalleFactura = ImpuestoXDetalleFactura.IdDetalleFactura
    INNER JOIN Factura
    ON Factura.IdDetalleFactura = DetalleFactura.IdDetalleFactura
    WHERE IdFactura = @IdFactura

```

```

GO

```

```

CREATE PROCEDURE [dbo].[SeleccionarImpuestoXDetalleFactura](@IdImpuesto INT, @IdDetalleFactura INT)
AS
    SELECT ImpuestoXDetalleFactura.IdImpuesto,
    ImpuestoXDetalleFactura.IdDetalleFactura,Impuesto.Impuesto,DetalleFactura.SubTotal
    FROM ImpuestoXDetalleFactura inner join Impuesto on ImpuestoXDetalleFactura.IdImpuesto =
    Impuesto.IdImpuesto
    inner join DetalleFactura on ImpuestoXDetalleFactura.IdDetalleFactura =
    DetalleFactura.IdDetalleFactura
    WHERE ImpuestoXDetalleFactura.IdImpuesto = ISNULL(@IdImpuesto,
    ImpuestoXDetalleFactura.IdImpuesto) AND
    ImpuestoXDetalleFactura.IdDetalleFactura = ISNULL(@IdDetalleFactura,
    ImpuestoXDetalleFactura.IdDetalleFactura)
GO

```

```

CREATE PROCEDURE [dbo].[SeleccionarInventario]
    @IdInventario int, @NombreInventario varchar(50), @Fecha date, @IdSucursal int
AS
BEGIN
    SELECT IdInventario,NombreInventario,Fecha,Inventario.IdSucursal,Sucursal.Nombre
    FROM Inventario inner join Sucursal on Inventario.IdSucursal = Sucursal.IdSucursal
    WHERE IdInventario=ISNULL(@IdInventario,IdInventario) and
    NombreInventario=ISNULL(@NombreInventario,NombreInventario) and
    Fecha=ISNULL(@Fecha,Fecha) and
    Inventario.IdSucursal = isnull(@IdSucursal,Inventario.IdSucursal)
END
GO

```

```

create procedure [dbo].[seleccionarPais]
    @Nombre varchar(50)
AS

```

```

BEGIN

    SELECT Nombre,IdPais,Locacion
    FROM Pais
    WHERE Nombre = isnull(@Nombre,Pais.Nombre)
END
GO

CREATE PROCEDURE [dbo].[SeleccionarPedido]
    @IdPedido int, @EstatusPedido Varchar(50), @FechaPedido Date, @IdEmpleadoPedido int, @IdFabrica
int
AS
BEGIN
    SELECT
    Pedido.IdPedido,Pedido.EstatusPedido,Pedido.FechaPedido,Pedido.IdEmpleadoPedido,Pedido.IdFabrica,Empleado
.Nombre as NombreEmpleado,Empleado.Apellido as EmpleadoApellido,Empleado.Cedula as EmpleadoCedula,
    Fabrica.Descripcion as Fabrica
    FROM Pedido inner join Fabrica on Pedido.IdFabrica = Fabrica.IdFabrica
    inner join Empleado on Pedido.IdEmpleadoPedido = Empleado.IdEmpleado
    WHERE
        Pedido.IdPedido=isnull(@IdPedido,Pedido.IdPedido) AND
        Pedido.EstatusPedido = isnull(@EstatusPedido,Pedido.EstatusPedido) AND
        Pedido.FechaPedido = isnull(@FechaPedido,Pedido.FechaPedido) AND
        Pedido.IdEmpleadoPedido=ISNULL(@IdEmpleadoPedido,Pedido.IdEmpleadoPedido)
AND
        Pedido.IdFabrica=ISNULL(@IdFabrica,Pedido.IdFabrica)
END
GO

CREATE PROCEDURE [dbo].[SeleccionarPedidoXVehiculo]
    @IdPedidoXVehiculo int, @IdPedido int, @IdVehiculo int, @FechaCompletado date
AS
BEGIN
    SELECT
    PedidoXVehiculo.IdPedidoXVehiculo,PedidoXVehiculo.IdPedido,PedidoXVehiculo.IdVehiculo,PedidoXVehiculo.Fech
aCompletado,Vehiculo.Marca as Marca,Vehiculo.Modelo as Modelo,Pedido.EstatusPedido as Estatus
    FROM PedidoXVehiculo inner join Vehiculo on PedidoXVehiculo.IdVehiculo = Vehiculo.IdVehiculo
    inner join Pedido on PedidoXVehiculo.IdPedido = Pedido.IdPedido
    WHERE
    PedidoXVehiculo.IdPedidoXVehiculo=ISNULL(@IdPedidoXVehiculo,PedidoXVehiculo.IdPedidoXVehiculo) AND
        PedidoXVehiculo.IdPedido = isnull(@IdPedido,PedidoXVehiculo.IdPedido) AND
        PedidoXVehiculo.IdVehiculo = isnull(@IdVehiculo,PedidoXVehiculo.IdVehiculo) AND

    PedidoXVehiculo.FechaCompletado=ISNULL(@FechaCompletado,PedidoXVehiculo.FechaCompletado)
END
GO

create procedure [dbo].[seleccionarProvincia]
    @Nombre varchar(50),
    @Pais varchar(50)
AS

```

```

BEGIN

    SELECT
Provincia.Nombre,Provincia.IdProvincia,Provincia.LocacionProvincia,Provincia.IdPais,Pais.Nombre
    FROM Pais inner join Provincia on Pais.IdPais = Provincia.IdPais
    WHERE Provincia.Nombre = isnull(@Nombre,Provincia.Nombre) AND Pais.Nombre =
isnull(@Pais,Pais.Nombre)
END
GO

create procedure [dbo].[seleccionarPuestoEmpleado]
    @Puesto varchar(50)

AS
BEGIN
    SELECT Puesto,IdPuestoEmpleado
    FROM PuestoEmpleado
    WHERE Puesto = isnull(@Puesto,Puesto)
END
GO

create procedure [dbo].[seleccionarSucursal]
    @Nombre varchar(50),
    @Descripcion varchar(50),
    @IdUbicacion int,
    @NombrePais varchar(50),
    @NombreProvincia varchar(50)

AS
BEGIN

    SELECT
Sucursal.IdSucursal,Sucursal.Nombre,Sucursal.Descripcion,Ubicacion.Descripcion,Provincia.Nombre,Pais.Nombre
    FROM Sucursal INNER JOIN Ubicacion on Sucursal.IdUbicacion = Ubicacion.IdUbicacion,Pais,Provincia
    WHERE Ubicacion.IdProvincia = Provincia.IdProvincia AND Provincia.IdPais = Pais.IdPais AND
        Sucursal.Nombre = isnull(@Nombre,Sucursal.Nombre) AND Sucursal.Descripcion =
isnull(@Descripcion,Sucursal.Descripcion) AND
        Sucursal.IdUbicacion = isnull(@IdUbicacion,Sucursal.IdUbicacion) AND Provincia.Nombre =
ISNULL(@NombreProvincia,Provincia.Nombre) AND
        Pais.Nombre = ISNULL(@NombrePais,Pais.Nombre)

END
GO

CREATE PROCEDURE [dbo].[SeleccionarTipoPagos](@IdTipoPago INT)
AS
    SELECT Pago
    FROM TipoPago
    WHERE IdTipoPago = ISNULL(@IdTipoPago, IdTipoPago)
GO

```

```

create procedure [dbo].[seleccionarTipoVehiculo]
    @Nombre varchar(50)
AS
BEGIN
    SELECT Nombre,IdTipoVehiculo
    FROM TipoVehiculo
    WHERE Nombre = isnull(@Nombre,Nombre)
END
GO

create procedure [dbo].[seleccionarUbicacion]
    @Descripcion varchar(50),
    @Provincia varchar(50)
AS
BEGIN
    SELECT
    Ubicacion.Descripcion,Ubicacion.IdProvincia,Ubicacion.LocacionExacta,Ubicacion.IdUbicacion,Provincia.Nombre
    FROM Provincia inner join Ubicacion on Provincia.IdProvincia = Ubicacion.IdUbicacion
    WHERE Ubicacion.Descripcion = isnull(@Descripcion,Ubicacion.Descripcion) AND Provincia.Nombre =
isnull(@Provincia,Provincia.Nombre)
END
GO

create procedure [dbo].[seleccionarVehiculo]
    @Tipo varchar(50),
    @Combustible varchar(50),
    @Marca varchar(50),
    @Modelo varchar(50),
    @PrecioLow int,
    @PrecioHigh int,
    @usado bit,
    @puertas int
AS
BEGIN
    SELECT Vehiculo.IdVehiculo,TipoVehiculo.Nombre as Tipo,Combustible.Nombre as
Combustible,Vehiculo.Marca,Vehiculo.Modelo,Vehiculo.Precio,Vehiculo.Puertas,Vehiculo.Usado
    FROM Vehiculo inner join TipoVehiculo on Vehiculo.IdTipoVehiculo = TipoVehiculo.IdTipoVehiculo
    inner join Combustible on Vehiculo.IdCombustible = Combustible.IdCombustible
    WHERE Vehiculo.Marca = isnull(@Marca,Vehiculo.Marca) AND Vehiculo.Modelo =
isnull(@Modelo,Vehiculo.Modelo) AND Vehiculo.Precio between isnull(@PrecioLow,Vehiculo.Precio) and
isnull(@PrecioHigh,Vehiculo.Precio)
    AND TipoVehiculo.Nombre = isnull(@Tipo,TipoVehiculo.Nombre) AND Combustible.Nombre =
isnull(@Combustible,Combustible.Nombre) and
    Vehiculo.Usado = ISNULL(@usado, Vehiculo.Usado) AND Vehiculo.Puertas =
isnull(@puertas,Vehiculo.Puertas)
END
GO

```

```

create procedure [dbo].[seleccionarVehiculoComprado]
    @Tipo varchar(50),
    @Combustible varchar(50),
    @Marca varchar(50),
    @Modelo varchar(50),
    @PrecioLow int,
    @PrecioHigh int
AS
BEGIN

    SELECT Vehiculo.IdVehiculo,TipoVehiculo.Nombre as Tipo,Combustible.Nombre as
    Combustible,Vehiculo.Marca,Vehiculo.Modelo,Vehiculo.Precio
    FROM Vehiculo inner join TipoVehiculo on Vehiculo.IdTipoVehiculo = TipoVehiculo.IdTipoVehiculo
        inner join Combustible on Vehiculo.IdCombustible = Combustible.IdCombustible
        inner join FacturaXVehiculo on FacturaXVehiculo.IdVehiculo = Vehiculo.IdVehiculo
    WHERE Vehiculo.Marca = isnull(@Marca,Vehiculo.Marca) AND Vehiculo.Modelo =
    isnull(@Modelo,Vehiculo.Modelo) AND Vehiculo.Precio between isnull(@PrecioLow,Vehiculo.Precio) and
    isnull(@PrecioHigh,Vehiculo.Precio)
        AND TipoVehiculo.Nombre = isnull(@Tipo,TipoVehiculo.Nombre) AND Combustible.Nombre =
    isnull(@Combustible,Combustible.Nombre)
END
GO

CREATE PROCEDURE [dbo].[SeleccionarVehiculoFabrica]
    @IdVehiculoFabrica int, @IdVehiculo int, @CostoVehiculo int, @IdInventario int, @IdFabrica int
AS
BEGIN
    SELECT
    VehiculoFabrica.IdVehiculoFabrica,VehiculoFabrica.IdVehiculo,VehiculoFabrica.CostoVehiculo,VehiculoFabrica.IdInv
    entario,VehiculoFabrica.IdFabrica,Fabrica.Descripcion as Fabrica,
        Vehiculo.Modelo as Modelo,Vehiculo.Marca as Marca
    FROM VehiculoFabrica inner join Fabrica on VehiculoFabrica.IdFabrica = Fabrica.IdFabrica
        inner join Vehiculo on VehiculoFabrica.IdVehiculo = Vehiculo.IdVehiculo
        inner join Inventario on VehiculoFabrica.IdInventario = Inventario.IdInventario
    WHERE
    VehiculoFabrica.IdVehiculoFabrica=ISNULL(@IdVehiculoFabrica,VehiculoFabrica.IdVehiculoFabrica) and
        VehiculoFabrica.IdVehiculo=ISNULL(@IdVehiculo,VehiculoFabrica.IdVehiculo) and

    VehiculoFabrica.CostoVehiculo=ISNULL(@CostoVehiculo,VehiculoFabrica.CostoVehiculo) and
        VehiculoFabrica.IdInventario=Isnull(@IdInventario,VehiculoFabrica.IdInventario) and
        VehiculoFabrica.IdFabrica = isnull(@IdFabrica,VehiculoFabrica.IdFabrica)
END
GO

CREATE PROCEDURE [dbo].[SeleccionarVehiculoFabricaXEmpleado]
    @IdVehiculoFabricaXEmpleado int,@IdVehiculoFabrica int, @IdEmpleado int, @Labor Varchar(50),
    @FechaInicio Date, @FechaFinal Date,@Estatus Varchar(50)
AS
BEGIN

```

```

SELECT
IdVehiculoFabricaXEmpleado,IdVehiculoFabrica,VehiculoFabricaXEmpleado.IdEmpleado,Labor,FechaInicio,FechaFi
nal,Estatus,Empleado.Nombre as Nombre, Empleado.Apellido as Apellido,
        Empleado.Cedula as Cedula
FROM VehiculoFabricaXEmpleado inner join Empleado on VehiculoFabricaXEmpleado.IdEmpleado =
Empleado.IdEmpleado
WHERE
IdVehiculoFabricaXEmpleado=ISNULL(@IdVehiculoFabricaXEmpleado,IdVehiculoFabricaXEmpleado) and
        IdVehiculoFabrica=ISNULL(@IdVehiculoFabrica,IdVehiculoFabrica)and

VehiculoFabricaXEmpleado.IdEmpleado=ISNULL(@IdEmpleado,VehiculoFabricaXEmpleado.IdEmpleado)and
        Labor=Isnull(@Labor,Labor)and
        FechaInicio=ISNULL(@FechaInicio,FechaInicio)and
        FechaFinal=ISNULL(@FechaFinal,FechaFinal)and
        Estatus=ISNULL(@Estatus,Estatus)

END
GO

create procedure [dbo].[seleccionarVehiculoXSucursal]
        @idVehiculo int,
        @idSucursal int

AS
BEGIN

        SELECT Vehiculo.Marca,Vehiculo.Modelo,Sucursal.Nombre,VehiculoXSucursal.IdVehiculoXSucursal
FROM   VehiculoXSucursal inner join Vehiculo on VehiculoXSucursal.IdVehiculo = Vehiculo.IdVehiculo
        inner join Sucursal on VehiculoXSucursal.IdSucursal = Sucursal.IdSucursal
WHERE  VehiculoXSucursal.IdSucursal = isnull(@idSucursal,VehiculoXSucursal.IdSucursal) AND
VehiculoXSucursal.IdVehiculo = isnull(@IdVehiculo,VehiculoXSucursal.IdVehiculo)
END
GO

/*****Fin script de SPs*****/

```

7.3. Scripts de Llenado de la base

```

USE Proyecto_BasesII;
GO

DECLARE @fecha DATETIME;
SET @fecha = GETDATE();

INSERT INTO TipoModalidad(Nombre)
VALUES('Credito'),('Contado');

INSERT INTO Usuario(Usuario, Pass, Privilegio)
VALUES('Usuario', 'Pass', 'Cliente'),
('Administrador', 'Pass', 'Administrador')

EXEC insertarPais 'Costa Rica', 'LINESTRING (100 100, 20 180, 180 180)'

```

```

EXEC insertarPais 'Argentina', 'LINESTRING (100 100, 20 180, 180 180)'
EXEC insertarPais 'Venezuela', 'LINESTRING (100 100, 20 180, 180 180)'

EXEC insertarProvincia 'Alajuela', 'LINESTRING (100 100, 20 180, 180 180)', 1
EXEC insertarProvincia 'Cartago', 'LINESTRING (100 100, 20 180, 180 180)', 2
EXEC insertarProvincia 'Heredia', 'LINESTRING (100 100, 20 180, 180 180)', 3

EXEC insertarUbicacion 'Casa 24F', 'POINT(2 2)', 1
EXEC insertarUbicacion 'Casa 25F', 'POINT(4 2)', 2
EXEC insertarUbicacion 'Casa 26F', 'POINT(9 2)', 3
EXEC insertarUbicacion 'Casa 27F', 'POINT(2 6)', 1
EXEC insertarUbicacion 'Casa 28F', 'POINT(4 10)', 2
EXEC insertarUbicacion 'Casa 29F', 'POINT(9 20)', 3

EXEC insertarCaracteristica 'Spoiler1', 'Rojo'
EXEC insertarCaracteristica 'Spoiler2', 'Azul'
EXEC insertarCaracteristica 'Spoiler3', 'Verde'

EXEC insertarExtra 'Nombre 1', 'Descripcion 1', 150000
EXEC insertarExtra 'Nombre 2', 'Descripcion 2', 250000
EXEC insertarExtra 'Nombre 3', 'Descripcion 3', 350000

EXEC insertarTipoVehiculo '4X4'
EXEC insertarTipoVehiculo 'Convertible'
EXEC insertarTipoVehiculo '2X2'

EXEC insertarCombustible 'Diesel'
EXEC insertarCombustible 'Super'
EXEC insertarCombustible 'Hibrido'

EXEC insertarVehiculo '4X4', 'Super', 'Ford', 'Mustang', 10000000
EXEC insertarVehiculo 'Convertible', 'Hibrido', 'Hyundai', 'Mustang', 100
EXEC insertarVehiculo 'Convertible', 'Hibrido', 'Honda', '2009', 100

EXEC insertarFotoVehiculo 'C:\Users\pierr\Pictures\imagen.jpg', 'Nombre', @fecha, 1
EXEC insertarFotoVehiculo 'C:\Users\pierr\Pictures\imagen.jpg', 'Nombre', @fecha, 2

EXEC insertarExtraXVehiculo 1, 1, 150000
EXEC insertarExtraXVehiculo 2, 1, 250000
EXEC insertarExtraXVehiculo 3, 2, 350000

EXEC insertarCaracteristicaXVehiculo 1, 1
EXEC insertarCaracteristicaXVehiculo 2, 1
EXEC insertarCaracteristicaXVehiculo 3, 2

EXEC insertarSucursal 'Nombre Sucursal 1', 'Una sucursal', 1
EXEC insertarSucursal 'Nombre Sucursal 2', 'Una sucursal', 2
EXEC insertarSucursal 'Nombre Sucursal 3', 'Una sucursal', 3

EXEC insertarPuestoEmpleado 'Supervisor'

```

EXEC insertarPuestoEmpleado 'Vendedor'
EXEC insertarPuestoEmpleado 'Facturador'

INSERT INTO Empleado(Nombre, Apellido, Telefono, Correo, IdSupervisor, IdPuestoEmpleado, IdSucursal, Cedula, IdUsuario)
VALUES ('Supervisor', 'Supervisor', '1111-1111', 'supervisor@gmail.com', null, 1, 1, '1', 2);

EXEC insertarEmpleado 'Gabriel', 'Piedra', '2121-2121', 'usuario@gmail.com', '1', 'Vendedor', 1, '2', 2
EXEC insertarEmpleado 'Luis', 'Piedra', '2121-2121', 'usuario@gmail.com', '1', 'Vendedor', 1, '3', 2
EXEC insertarEmpleado 'Roberto', 'Piedra', '2121-2121', 'usuario@gmail.com', '1', 'Vendedor', 1, '4', 2

EXEC InsertarCliente 1, 'Israel', 'Piedra', '1111-1111', 'cliente@gmail', 4
EXEC InsertarCliente 1, 'Jean', 'Piedra', '1111-1111', 'cliente@gmail', 5
EXEC InsertarCliente 1, 'Pablo', 'Piedra', '1111-1111', 'cliente@gmail', 6

EXEC InsertarDescuento 'Descuento de verano', 20
EXEC InsertarDescuento 'Descuento de verano', 60
EXEC InsertarDescuento 'Descuento de verano', 40

EXEC InsertarImpuesto 'Impuesto', 2
EXEC InsertarImpuesto 'Impuesto', 4
EXEC InsertarImpuesto 'Impuesto', 6

EXEC InsertarTipoPago 'Efectivo'
EXEC InsertarTipoPago 'Tarjeta'
EXEC InsertarTipoPago 'Cheque'

EXEC InsertarDetalleFactura 1, 1, 12000, 1, 'Comentario', 1
EXEC InsertarDetalleFactura 2, 2, 14000, 2, 'Comentario', 2

EXEC InsertarFactura '1', 1, 150000, @fecha
EXEC InsertarFactura '2', 2, 250000, @fecha

EXEC InsertarDescuentoXDetalleFactura 1, 1

EXEC InsertarImpuestoXDetalleFactura 1, 1

EXEC InsertarFacturaXVehiculo 1, 1
EXEC InsertarFacturaXVehiculo 2, 2

EXEC InsertarComision 1, 1, 25
EXEC InsertarComision 2, 2, 50

EXEC InsertarEntrega 1, 4, 'En proceso', @fecha
EXEC InsertarEntrega 2, 5, 'Enviado', @fecha

EXEC InsertarFabrica 1, 'Una fabrica'
EXEC InsertarFabrica 2, 'Una fabrica'

EXEC InsertarPedido 'En proceso', @fecha, 1, 1

EXEC InsertarPedido 'Entregado', @fecha, 2, 2
EXEC InsertarPedido 'Cancelado', @fecha, 3, 2

EXEC InsertarPedidoXVehiculo 1, 1, @fecha
EXEC InsertarPedidoXVehiculo 2, 2, @fecha

EXEC InsertarInventario 'Inventario', @fecha, 1
EXEC InsertarInventario 'Inventario', @fecha, 2

EXEC InsertarVehiculoFabrica 1, 150000, 1, 1
EXEC InsertarVehiculoFabrica 2, 450000, 2, 2

EXEC InsertarVehiculoFabricaXEmpleado 1, 3, 'Montar el chasis', @fecha, @fecha, 'Progreso'
EXEC InsertarVehiculoFabricaXEmpleado 2, 3, 'Montar las llantas', @fecha, @fecha, 'Completado'

INSERT INTO VehiculoXSucursal(IdVehiculo, IdSucursal)
VALUES(1, 1),
(2, 2),
(3, 3)

/*****Fin de scripts de llenado de datos*****/