

**UNIVERSIDAD BETHESDA  
CARRERA DE INGENIERÍA INFORMÁTICA**



**BASE DE DATOS DE UN COTEXTO ACADEMICO DE AREA  
ACADEMICA “UNIBETH”**

**Universitari@:**

- Rodrigo Agustin Flores Via

**Lic:** Tito Flores

**Proyecto de Innovación Creatividad:**

Asignatura Base de Datos II

**Santa Cruz de la Sierra, Bolivia 2023**

## Tabla de contenido

<b>1. Acercamiento al contexto de negocio</b>	<b>3</b>
<b>2. Enunciado de problema</b>	<b>3</b>
<b>3. Diseño de Base de Datos</b>	<b>3</b>
3.1. Actores de negocio	3
3.2. Entidades de Negocio	4
3.3. Relaciones	3
.....	7
3.4. Mapeo	8
3.5. Tabla de Volumen	5
.....	9
3.6. Diagrama de Clases	15
<b>4. Implementación de Base de Datos</b>	<b>16</b>
4.1. Diseño Físico	16
4.2. Modelo Físico	16
4.3. Procedimientos Almacenados	28
4.4. Función	29
4.5. Trigger	29
<b>5. Herramienta de Desarrollo</b>	<b>29</b>
5.1. Software	29
5.2. Hardware	30
<b>6. Anexo</b>	<b>30</b>

## 1. Acercamiento al contexto de negocio

La empresa "CONSUTRANS" provee servicios de transporte de personal, alquiler de vehículos y dotación de conductores se ubica en [5to anillo y Radial 17 1/2 Santa Cruz, Andrés Ibáñez, BO] y fue fundada [en el año 2005] con el objeto de proveer transporte de personas y alquiler de vehículos.

## 2. Enunciado de problema

La empresa "CONSUTRANS" se dedica al alquiler de vehículos y servicios de transporte. La empresa necesita una base de datos que le permita gestionar eficientemente sus operaciones, incluyendo el alquiler de vehículos y la gestión de clientes y conductores. Se desea desarrollar un sistema para gestionar el alquiler de vehículos. El sistema debe mantener información sobre personas, clientes, conductores, vehículos, alquileres, mantenimiento de vehículos, reservas, facturación, pagos, empleados y seguros. Cada entidad tiene atributos específicos y relaciones con otras entidades.

## 3. Diseño de Base de Datos

### 3.1 Actores de Negocio

- **Persona**

Atributo	Tipo de datos
ID Persona	Números
Nombre	Texto
Apellido	Texto
Teléfono	Números
Correo electrónico	Texto
Dirección	Números

- **Clientes**

Atributo	Tipo de datos
ID Cliente	Números
ID Persona	Números

- **Conductores**

Atributo	Tipo de datos
ID Conductor	Números
ID Persona	Números
Licencia de Conducir	Texto

### 3.2 Entidades de Negocio

- **Vehículos**

Atributo	Tipo de Datos
ID Vehículo	Números
Tipo de Vehículo	Texto
Marca	Texto
Año	Números
Placa	Texto
Precio Por Alquiler Por Dia	Números

- **Alquileres**

Atributo	Tipo de datos
ID Alquiler	Números
ID Cliente	Números
ID Conductor	Números
ID Vehículo	Números
Fecha de Inicio Alquiler	Números
Fecha de Finalización Alquiler	Números
Total, a Pagar	Números

- **Mantenimiento de Vehículos**

Atributo	Tipo de datos
ID de Mantenimiento	Números
Fecha de Mantenimiento	Números
Tipo de Mantenimiento	Números
Costo	Números
ID Vehículo	Números

- **Facturación**

Atributo	Tipo de datos
ID Factura	Números
Fecha de Factura	Números
Monto	Números
Estado de Pago	Texto
ID Cliente	Números

- **Pagos**

Atributo	Tipo de datos
ID Pago	Números
Fecha de Pago	Números
Monto Pagado	Números
Métodos de Pago	Texto
ID Factura	Números

- **Empleados**

Atributo	Tipo de datos
ID Empleado	Números
ID Persona	Números
Cargo	Texto

- **Seguros**

Atributo	Tipo de datos
Numero de Póliza	Números
Compañías de Seguro	Números
Cobertura	Números
Fecha de Inicio de la Póliza	Números
Fecha de Vencimiento Póliza	Números
ID Vehículos	Números

### 3.3 Relaciones

Entidad	Relación	Entidad
Cliente	Es una	Persona
Conductor	Es una	Persona
Vehículo	Tiene una	Facturación
Factura	Está sujeta a	Pago
Vehículo	Tiene	Mantenimiento
Empleado	Es una	Persona

### 3.4 Mapeo

#### 2.4. Mapeo

- **Persona**

<b>Id Persona</b>	<b>Nombre</b>	<b>Apellido</b>	<b>Teléfono</b>	<b>Correo electrónico</b>	<b>Dirección</b>
PK					

- **Clientes**

<b>Id Cliente</b>	<b>Id Persona</b>
PK	FK

- **Conductores**

<b>Id Conductor</b>	<b>Id Persona</b>	<b>Licencia de conducir</b>
PK	FK	

- **Vehículos**

<b>Id de Vehículo</b>	<b>Tipo de Vehículo</b>	<b>Marca</b>	<b>Año</b>	<b>Placa</b>	<b>Precio de alquiler por día</b>
PK					

- **Alquileres**

<b>Id de Alquiler</b>	<b>Id Cliente</b>	<b>Id Conductor</b>	<b>Id de Vehículo</b>	<b>Fecha de inicio del alquiler</b>	<b>Fecha de finalización de alquiler</b>	<b>Total a pagar</b>
PK	FK	FK	FK			

- **Mantenimiento de Vehículos**

<b>Id de Mantenimiento</b>	<b>Fecha de Mantenimiento</b>	<b>Tipo de Mantenimiento</b>	<b>Costo</b>	<b>Id de Vehículo</b>
PK				FK



- **Reservas**

<b>Id de Reserva</b>	<b>Fecha de Reserva</b>	<b>Fecha de Inicio del Alquiler</b>	<b>Fecha de Finalización del Alquiler</b>	<b>Id Vehículo</b>	<b>Id Cliente</b>
PK				FK	FK

- **Facturación**

<b>Id de Factura</b>	<b>Fecha de Factura</b>	<b>Monto</b>	<b>Estado de Pago</b>	<b>Id Cliente</b>
PK				FK

- **Pagos**

<b>Id de Pago</b>	<b>Fecha de Pago</b>	<b>Monto Pagado</b>	<b>Métodos de Pago</b>	<b>Id de Factura</b>
PK				FK

- **Empleados**

<b>Id Empleado</b>	<b>Id Persona</b>	<b>Cargo</b>
PK	FK	

- **Seguros**

<b>Numero de Póliza</b>	<b>Compañía de Seguros</b>	<b>Cobertura</b>	<b>Fecha de Inicio de Póliza</b>	<b>Fecha de Vencimiento de la Póliza</b>	<b>Id Vehículo</b>
PK					FK

### 3.5 Tabla de volumen

- **Persona**

<b>ATRIBUTOS</b>	<b>LLAVES</b>	<b>TIPO DE DATOS</b>	<b>AMPLITUD</b>	<b>NULO</b>	<b>DESCRIPCION</b>
<b>ID Persona</b>	<b>PK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Identificación única de la persona</b>

<b>Nombre</b>		<b>Varchar</b>	<b>50</b>	<b>NO</b>	<b>Nombre de la persona</b>
<b>Apellido</b>		<b>Varchar</b>	<b>50</b>	<b>NO</b>	<b>Apellido de la persona</b>
<b>Telefono</b>		<b>Varchar</b>	<b>15</b>	<b>SI</b>	<b>Número de teléfono</b>
<b>Correo electronico</b>		<b>Varchar</b>	<b>100</b>	<b>SI</b>	<b>Dirección de correo electrónico</b>
<b>Dirección</b>		<b>Varchar</b>	<b>100</b>	<b>SI</b>	<b>Dirección de la persona</b>

- Clientes**

<b>ATRIBUTOS</b>	<b>LLAVES</b>	<b>TIPO DE DATOS</b>	<b>AMPLITUD</b>	<b>NULO</b>	<b>DESCRIPCION</b>
<b>ID Cliente</b>	<b>PK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Identificación única del cliente</b>
<b>ID Persona</b>	<b>FK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Clave foránea hacia la tabla "Persona"</b>

- Conductores**

<b>ATRIBUTOS</b>	<b>LLAVES</b>	<b>TIPO DE DATOS</b>	<b>AMPLITUD</b>	<b>NULO</b>	<b>DESCRIPCION</b>
<b>ID Conductor</b>	<b>PK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Identificación única del conductor</b>
<b>ID Persona</b>	<b>FK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Clave foránea hacia la tabla "Persona"</b>
<b>Licencia de conducir</b>		<b>Varchar</b>	<b>20</b>	<b>SI</b>	<b>Número de licencia de conducir del conductor</b>

- Vehículos

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
ID Vehículo	PK	Int	1	NO	Identificación única del vehículo
Tipo Vehículo		Varchar	50	SI	Tipo de vehículo
Marca		Varchar	50	SI	Marca del vehículo
Año		Int	4	SI	Año del vehículo
Placa		Varchar	10	SI	Número de placa del vehículo
Precio día		Decimal	10,2	SI	Precio de alquiler por día

- Alquileres

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
ID Alquiler	PK	Int	1	NO	Identificación única del alquiler
ID Cliente	FK	Int	1	NO	Clave foránea hacia la tabla "Clientes"
ID Conductor	FK	Int	1	NO	Clave foránea hacia la tabla "Conductores"
ID Vehículo	FK	Int	1	NO	Clave foránea hacia la tabla "Vehículos"
Fecha Inicio Alquiler		Date		NO	Fecha de inicio del alquiler
Fecha Finalización Alquiler		Date		NO	Fecha de finalización del alquiler
Total Pagar		Decimal	10,2	NO	Total a pagar

- **Mantenimiento de Vehículos**

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
<b>ID Mantenimiento</b>	<b>PK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Identificación única del mantenimiento</b>
<b>Fecha Mantenimiento</b>		<b>Date</b>		<b>NO</b>	<b>Fecha de mantenimiento</b>
<b>Tipo Mantenimiento</b>		<b>Varchar</b>	<b>50</b>	<b>NO</b>	<b>Tipo de mantenimiento</b>
<b>Costo</b>		<b>Decimal</b>	<b>10,2</b>	<b>NO</b>	<b>Costo del mantenimiento</b>
<b>ID Vehículo</b>	<b>FK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Clave foránea hacia la tabla "Vehículos"</b>

- **Reservas**

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
<b>ID Reserva</b>	<b>PK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Identificación única de la reserva</b>
<b>Fecha Reserva</b>		<b>Date</b>		<b>NO</b>	<b>Fecha de reserva</b>
<b>Fecha Inicio Alquiler</b>		<b>Date</b>		<b>NO</b>	<b>Fecha de inicio del alquiler reservado</b>
<b>Fecha Finalización Alquiler</b>		<b>Date</b>		<b>NO</b>	<b>Fecha de finalización del alquiler reservado</b>
<b>ID Vehículo</b>	<b>FK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Clave foránea hacia la tabla "Vehículos"</b>
<b>ID Cliente</b>	<b>FK</b>	<b>Int</b>	<b>1</b>	<b>NO</b>	<b>Clave foránea hacia la tabla "Clientes"</b>

- **Facturación**

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
ID Factura	PK	Int	1	NO	Identificación única de la factura
Fecha Factura		Date		NO	Fecha de emisión de la factura
Monto		Decimal	10,2	NO	Monto total de la factura
Estado de Pago		Varchar	15	NO	Estado de pago (pagada o pendiente)
ID Cliente	FK	Int	1	NO	Clave foránea hacia la tabla "Clientes"

- **Pagos**

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
ID Pago	PK	Int	1	NO	Identificación única del pago
Fecha Pago		Date		NO	Fecha de pago
Monto Pagado		Decimal	10,2	NO	Monto pagado
Método Pagado		Varchar	50	SI	Método de pago específico (opcional)
ID Factura	FK	Int	1	NO	Clave foránea hacia la tabla "Facturación"

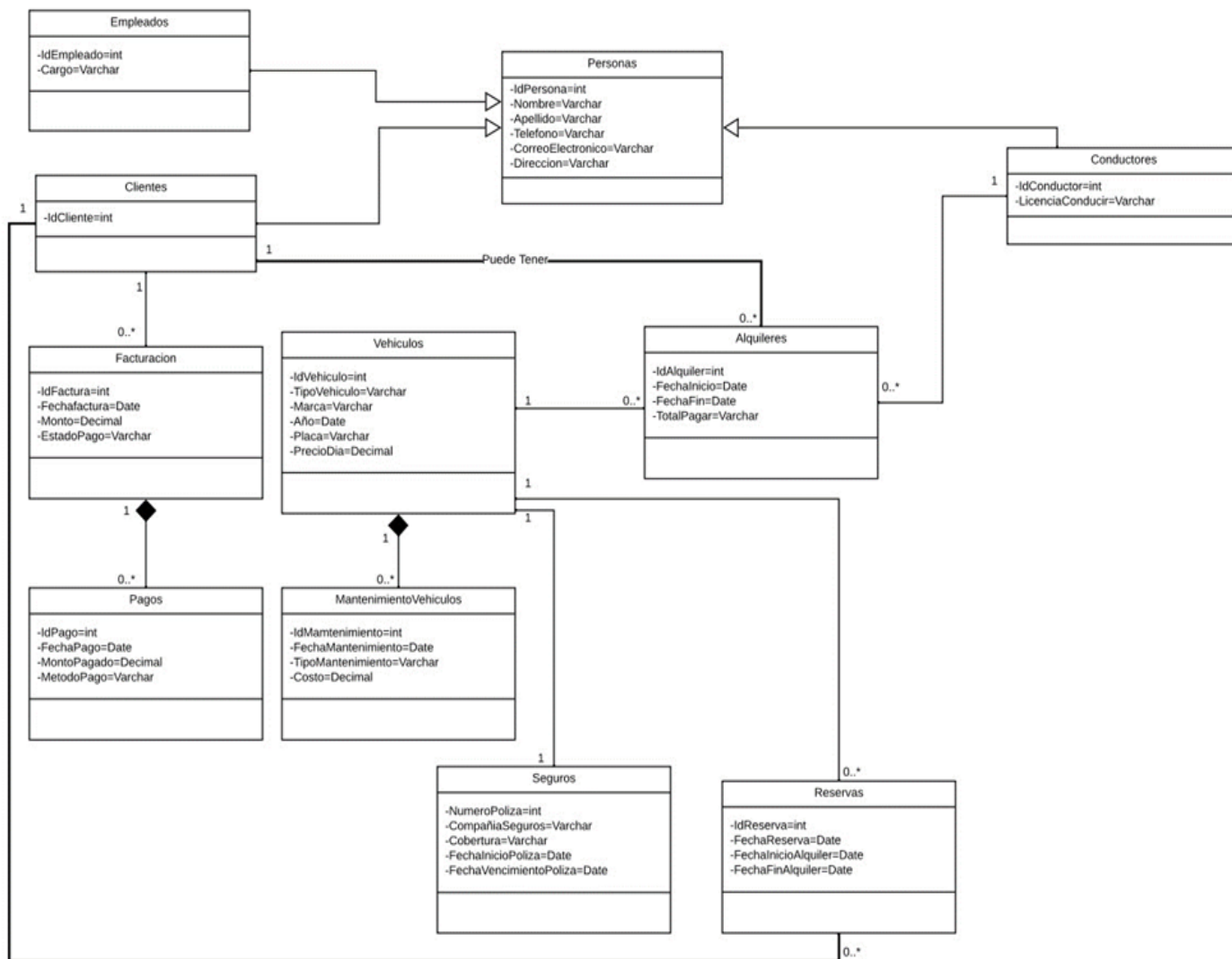
- Empleados

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
ID Empleado	PK	Int	1	NO	Identificación única del empleado
ID Persona	FK	Int	1	NO	Clave foránea hacia la tabla "Persona"
Cargo		Varchar	50	SI	Cargo del empleado

- Seguro

ATRIBUTOS	LLAVES	TIPO DE DATOS	AMPLITUD	NULO	DESCRIPCION
Numero Poliza	PK	Int	1	NO	Número de del póliza del seguro
Compañía Seguro		Varchar	50	NO	Compañía de seguros
Cobertura		Varchar	50	NO	Cobertura del seguro
Fecha Inicio Poliza		Date		NO	Fecha de inicio de la póliza
Fecha Vencimiento Poliza		Date		NO	Fecha de vencimiento de la póliza
ID Vehículo	FK	Int	1	NO	Clave foránea hacia la tabla "Vehículos"

### 3.6 Diagrama de Clases



**Gráfico:** Diagrama de Clases de Gestor Alquiler Vehículos

## 4.Implementación de Base de Datos

### 4.1 Diseño Físico

```
create database GestorAlquilerVehiculos;
```

```
use GestorAlquilerVehiculos;
```

```
-- crear la tabla de personas
```

```
create table Persona (
```

```
    IdPersona int auto_increment primary key,
```

```
    Nombre varchar(50) not null,
```

```
    Apellido varchar(50) not null,
```

```
    Telefono varchar(15),
```

```
    CorreoElectronico varchar(100),
```

```
    Direccion varchar(100)
```

```
);
```

```
-- crear la tabla de clientes
```

```
create table Clientes (
```

```
    IdCliente int auto_increment primary key,
```

```
    IdPersona int,
```

```
    foreign key (IdPersona) references Persona(IdPersona)
```

```
);
```



-- cree la tabla de conductores

```
create table Conductores (  
  
    IdConductor int auto_increment primary key,  
  
    IdPersona int,  
  
    foreign key (IdPersona) references Persona(IdPersona),  
  
    LicenciaConducir varchar(20)  
  
);
```

-- crear la tabla de vehiculos

```
create table Vehiculos (  
  
    IdVehiculo int auto_increment primary key,  
  
    TipoVehiculo varchar(50),  
  
    Marca varchar(50),  
  
    Año int,  
  
    Placa varchar(10),  
  
    PrecioAlquilerPorDia decimal(10, 2)  
  
);
```

-- crear la tabla de alquileres

```
create table Alquileres (  
  
    IdAlquiler int auto_increment primary key,
```

```
IdCliente int references Clientes(IdCliente),

IdConductor int references Conductores(IdConductor),

IdVehiculo int references Vehiculos(IdVehiculo),

FechaInicio date,

FechaFin date,

TotalPagar decimal(10, 2)

);

drop table alquileres;

-- crear la tabla de Mantenimiento de vehiculos

create table MantenimientoVehiculos (

    IdMantenimiento int auto_increment primary key,

    FechaMantenimiento date,

    TipoMantenimiento varchar(50),

    Costo decimal(10, 2),

    IdVehiculo int,

    foreign key (IdVehiculo) references Vehiculos(IdVehiculo)

);

drop table seguros;

-- crear la tabla de reservas
```

```
create table Reservas (  
  
    IdReserva int auto_increment primary key,  
  
    FechaReserva date,  
  
    FechaInicioAlquiler date,  
  
    FechaFinAlquiler date,  
  
    IdVehiculo int,  
  
    foreign key (IdVehiculo) references Vehiculos(IdVehiculo),  
  
    IdCliente int,  
  
    foreign key (IdCliente) references Clientes(IdCliente)  
  
);
```

-- crear la tabla de facturacion

```
create table Facturacion (  
  
    IdFactura int auto_increment primary key,  
  
    FechaFactura date,  
  
    Monto decimal(10, 2),  
  
    EstadoPago varchar(15),  
  
    IdCliente int,  
  
    foreign key (IdCliente) references Clientes(IdCliente)  
  
);
```

-- crear la tabla de pagos

create table Pagos (

IdPago int auto\_increment primary key,

FechaPago date,

MontoPagado decimal(10, 2),

MetodoPago varchar(50),

IdFactura int references Facturacion(IdFactura)

);

-- crear la tabla de empleados

create table Empleados (

IdEmpleado int auto\_increment primary key,

IdPersona int,

foreign key (IdPersona) references Persona(IdPersona),

Cargo varchar(50)

);

-- crear la tabla de seguros

create table Seguros (

NumeroPoliza int auto\_increment primary key,

CompañiaSeguros varchar(50),

```
Cobertura varchar(50),

FechaInicioPoliza date,

FechaVencimientoPoliza date,

IdVehiculo int,

foreign key (IdVehiculo) references Vehiculos(IdVehiculo)

);

-- insertamos datos a las tablas

-- insertamos Personas

insert into Persona ( Nombre, Apellido, Telefono, CorreoElectronico, Direccion)

values ( 'Carlos', 'Gutierrez', '7654321', 'carlos.gutierrez@gmail.com', 'Av. Monseñor Rivero'),

( 'Isabel', 'Lopez', '9876543', 'isabel.lopez@gmail.com', 'Calle Simon Bolivar'),

( 'Juan', 'Perez', '6783458', 'juan.perez@gmail.com', 'Calle Warnes'),

( 'Luis', 'Gonzales', '7682151', 'Luis.Gonzales@gmail.com', 'Av. Cañoto');

select * from persona;

-- Insertamos Clientes

insert into Clientes ( IdPersona)

values ( 1),

( 2),

( 3);
```

```
select * from clientes;
```

```
-- Insertamos Conductores
```

```
insert into Conductores ( IdPersona, LicenciaConducir)
```

```
values ( 4, 'SC789012');
```

```
select * from Conductores;
```

```
-- Insertamos Vehiculos
```

```
insert into Vehiculos ( TipoVehiculo, Marca, Año, Placa, PrecioAlquilerPorDia)
```

```
values ( 'Sedan', 'Toyota', 2022, 'SC-ABC123', 60.00),
```

```
( 'SUV', 'Honda', 2021, 'SC-XYZ789', 70.00);
```

```
select * from Vehiculos;
```

```
-- Insertamos Alquileres
```

```
insert into Alquileres ( IdCliente, IdConductor, IdVehiculo, FechaInicio, FechaFin)
```

```
values ( 1, 1, 2, '2023-01-01', '2023-01-10');
```

```
select * from Alquileres;
```

```
-- Insertamos MantenimientoVehiculo
```

```
insert into MantenimientoVehiculos ( FechaMantenimiento, TipoMantenimiento, Costo, IdVehiculo)
```

```
values ( '2023-03-01', 'Cambio de aceite', 40.00, 1),
```

```
( '2023-04-01', 'Reparación de frenos', 90.00, 2);
```

```
select * from MantenimientoVehiculos;
```

-- Insertamos Las Reservas

```
insert into Reservas ( FechaReserva, FechaInicioAlquiler, FechaFinAlquiler, IdVehiculo, IdCliente)
```

```
values( '2023-05-01', '2023-06-01', '2023-06-10', 1, 1);
```

```
select* from reservas;
```

-- Insertamos Facturacion

```
insert into Facturacion ( FechaFactura, Monto, EstadoPago, IdCliente)
```

```
values( '2023-01-10', 700.00, 'Pagado', 1);
```

```
insert into Facturacion ( FechaFactura, Monto, EstadoPago, IdCliente)
```

```
values( '2023-02-15', 500.00, 'Pendiente', 2);
```

```
insert into Facturacion ( FechaFactura, Monto, EstadoPago, IdCliente)
```

```
values( '2023-02-15', 300.00, 'Pendiente', 3);
```

```
select * from Facturacion;
```

-- Insertamos Pagos

```
Insert into Pagos ( FechaPago, MontoPagado, MetodoPago, IdFactura)
```

```
values ( '2023-01-10', 630.00, 'Tarjeta de crédito', 1);
```

```
insert into Pagos ( FechaPago, MontoPagado, MetodoPago, IdFactura)
```

```
values ( '2023-01-01', 500.00, 'Targeta de credito', 2);
```

```
select * from Pagos;
```

-- Insertamos Seguros

```
insert into Seguros ( CompaniaSeguros, Cobertura, FechaInicioPoliza, FechaVencimientoPoliza, IdVehiculo)

values ( 'BoliviaSeguros', 'Responsabilidad civil', '2023-01-01', '2024-01-01', 1),

( 'SantaCruzInsurance', 'Todo riesgo', '2023-02-01', '2024-02-01', 2);

select * from Seguros;

-- Cree este procedimiento para insertar un nuevo cliente

delimiter $$

create procedure InsertarCliente(

    in p_Nombre varchar(50),

    in p_Apellido varchar(50),

    in p_Telefono varchar(15),

    in p_CorreoElectronico varchar(100),

    in p_Direccion varchar(100)

)

begin

    insert into Persona(Nombre, Apellido, Telefono, CorreoElectronico, Direccion)

    values( p_Nombre, p_Apellido, p_Telefono, p_CorreoElectronico, p_Direccion);

end $$

call InsertarCliente('Eduardo', 'Ribera', '6438971', 'Eduardo.Ribera@gmail.com', 'Calle Florida');

select * from Persona;
```



-- Cree este procedimiento para calcular el total a pagar por alquiler

delimiter \$\$

DROP PROCEDURE IF EXISTS InsertarCliente;

create procedure CalcularTotalPagar(

in p\_IdAlquiler int

)

begin

declare p\_PrecioPorDia decimal(10, 2);

declare p\_DiasAlquiler int;

select PrecioAlquilerPorDia into p\_PrecioPorDia

from Alquileres

join Vehiculos on Alquileres.IdVehiculo = Vehiculos.IdVehiculo

where Alquileres.IdAlquiler = p\_IdAlquiler;

select DATEDIFF(FechaFin, FechaInicio) into p\_DiasAlquiler

from Alquileres

where IdAlquiler = p\_IdAlquiler;

```
update Alquileres
```

```
set TotalPagar = p_PrecioPorDia * p_DiasAlquiler
```

```
where IdAlquiler = p_IdAlquiler;
```

```
end $$
```

```
call CalcularTotalPagar(1);
```

```
select * from alquileres;
```

```
-- cree esta funcion para calcular el total pagado segun el monto pagado en la tabla pagos
```

```
delimiter $$
```

```
create function CalcularTotalPagado(
```

```
    idCliente int
```

```
) returns decimal(10, 2)
```

```
begin
```

```
    declare totalPagado decimal(10, 2);
```

```
-- Lógica para calcular el total pagado
```

```
select SUM(MontoPagado) into totalPagado
```

```
from Pagos
```

```
where IdCliente = idCliente;
```

```
        return totalPagado;

end $$

select CalcularTotalPagado(1) as TotalPagado;

select * from pagos;

-- cree este Trigger para actualizar el estado de pago en Facturacion después de realizar un pago en la tabla Pagos

delimiter $$

create trigger ActualizarEstadoPago

after insert on Pagos

for each row

begin

    update Facturacion

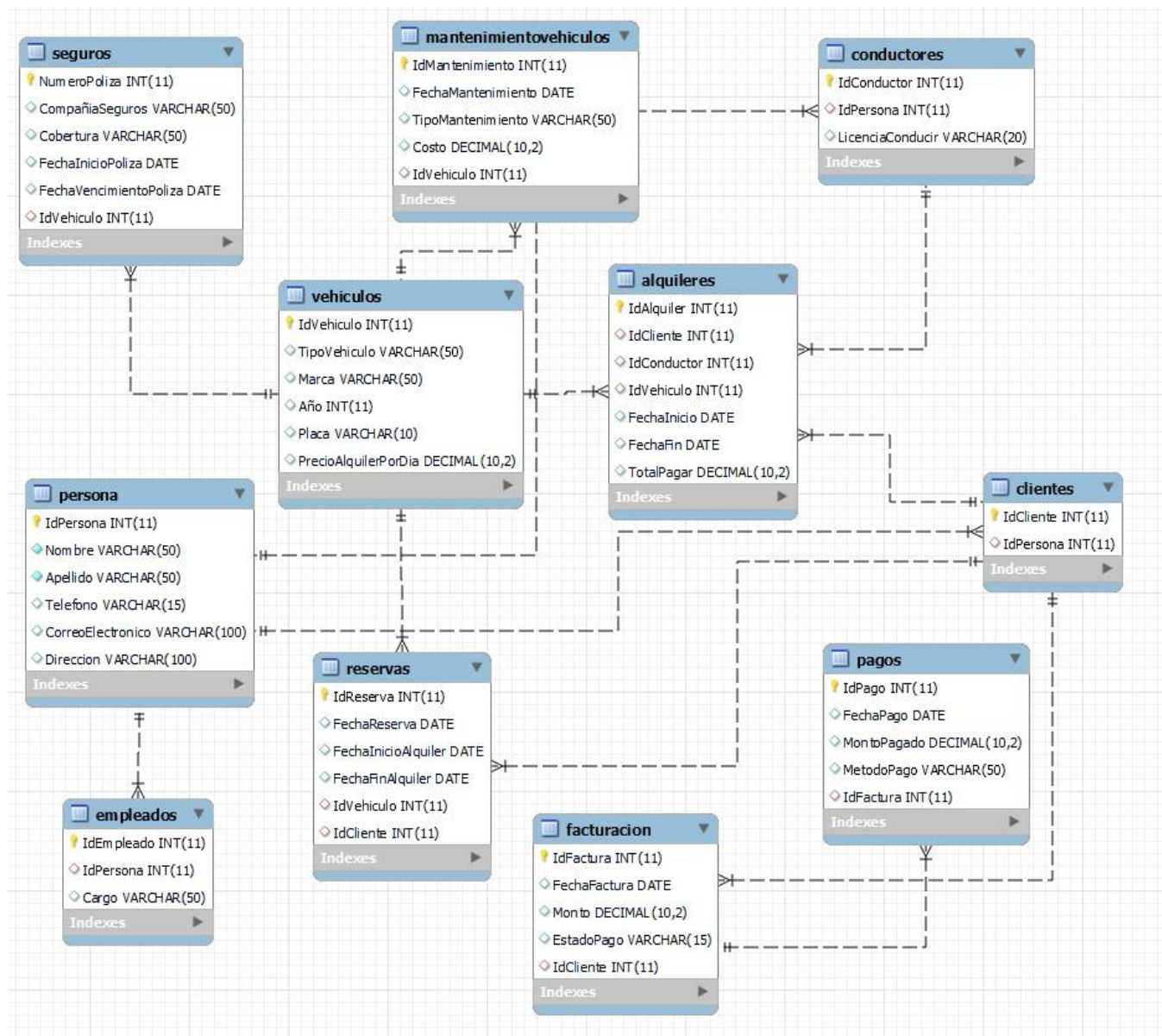
    set EstadoPago = 'Pagado'

    where IdFactura = new.IdFactura;

end $$

select * from Facturacion where IdFactura = 2;
```

## 4.2 Modelo Físico



## 4.3 Procedimientos Almacenados

Procedimiento	Descripción
InsertarCliente()	Cree este procedimiento almacenado para insertar un nuevo cliente a la tabla clientes
CalcularTotalPagar()	Cree este procedimiento almacenado para calcular el total a pagar por alquiler

#### 4.4 Función

Función	Descripción
CalcularTotalPagado()	Cree esta función para calcular el total pagado según el monto pagado en la tabla pagos

#### 4.5 Trigger

Trigger	Descripción
ActualizarEstadoPago	Cree este Trigger con el nombre ActualizarEstadoPago para actualizar el estado de pago en Facturación después de realizar un pago en la tabla Pagos

### 5.Herramientas de Desarrollo

#### 5.1 Software

➤ **Gestor de Base de Datos**

- **MYSQL Workbench y XAMP**

**MySQL Workbench:** es una herramienta de desarrollo visual para MySQL. Permite diseñar, modelar, generar y gestionar bases de datos de manera gráfica.

**XAMPP:** es un paquete que proporciona un entorno de desarrollo local completo para aplicaciones web. Incluye Apache como servidor web, MySQL como sistema de gestión de bases de datos, PHP como lenguaje de programación y Perl.

En este proyecto, utilicé MySQL Workbench como herramienta principal para el diseño y gestión de la base de datos. Aproveché su interfaz gráfica intuitiva para modelar la estructura de la base de datos, definiendo tablas y estableciendo relaciones de manera visual.

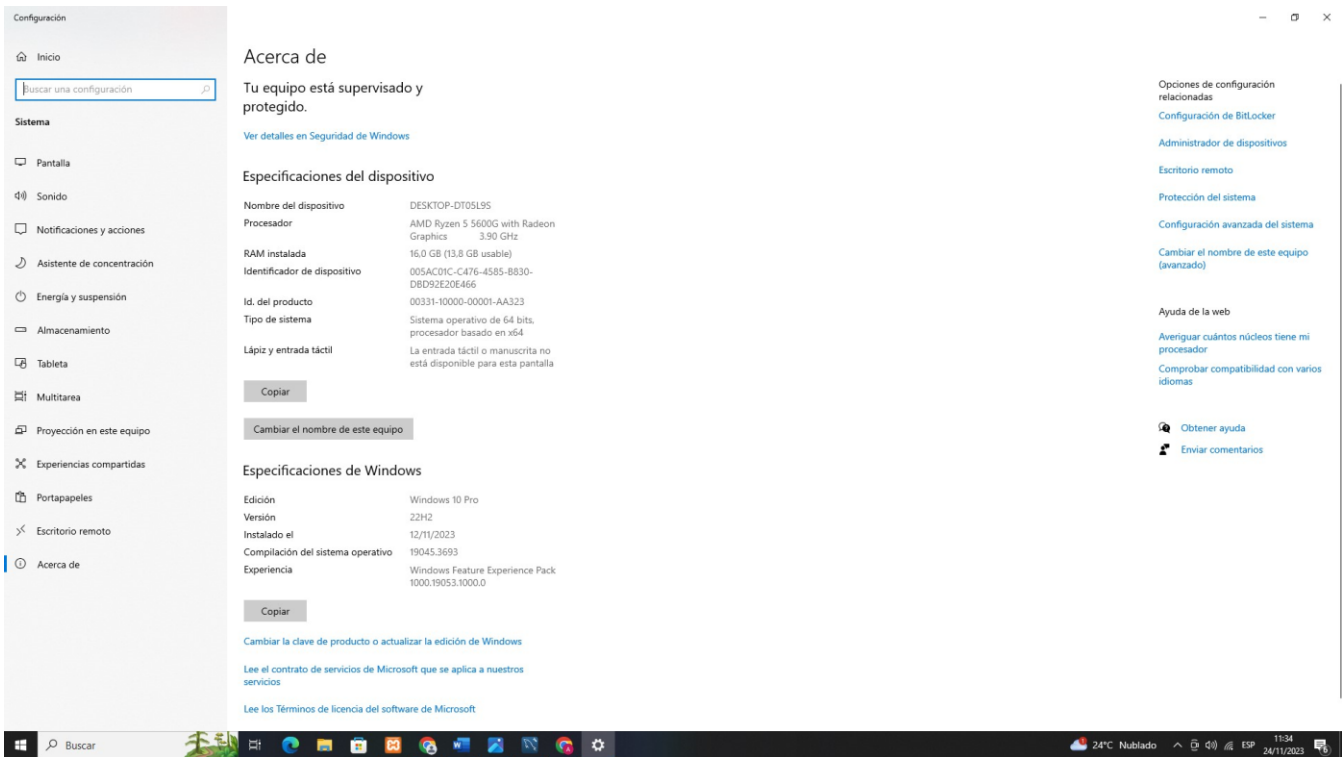
En cuanto a XAMPP, lo integré en mi entorno de desarrollo para disponer de un servidor web local y un sistema de gestión de bases de datos MySQL

➤ **Lenguaje de Programación**

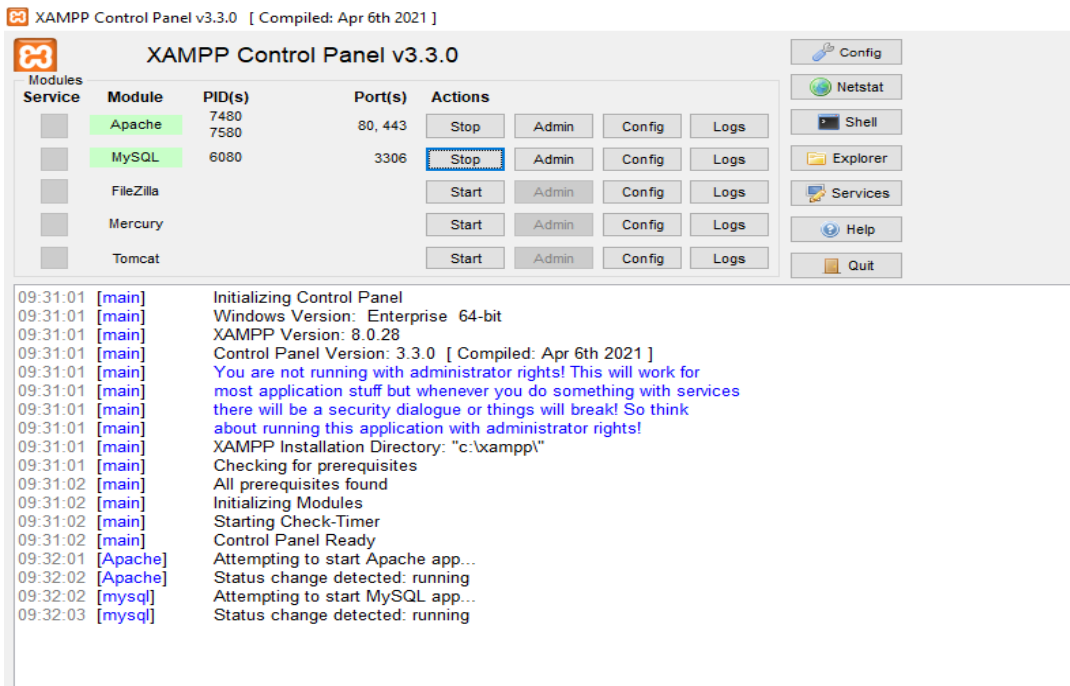
- **C# Visual Estudio:** es un lenguaje de programación desarrollado por Microsoft como parte de su plataforma .NET. Es un lenguaje orientado a objetos diseñado para ser simple, eficiente y seguro. Se utiliza comúnmente para el desarrollo de aplicaciones de escritorio, web y móviles en el entorno de desarrollo integrado (IDE) llamado Visual Studio.

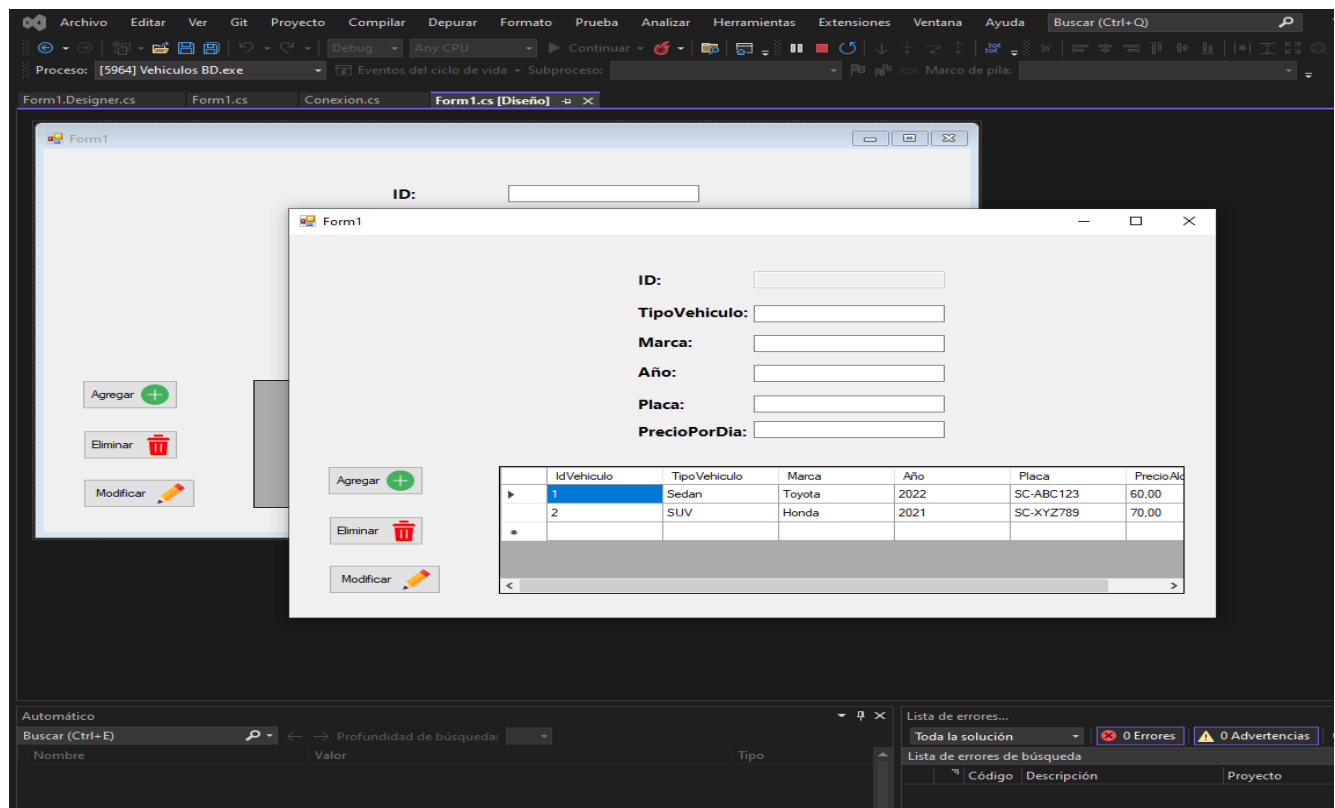
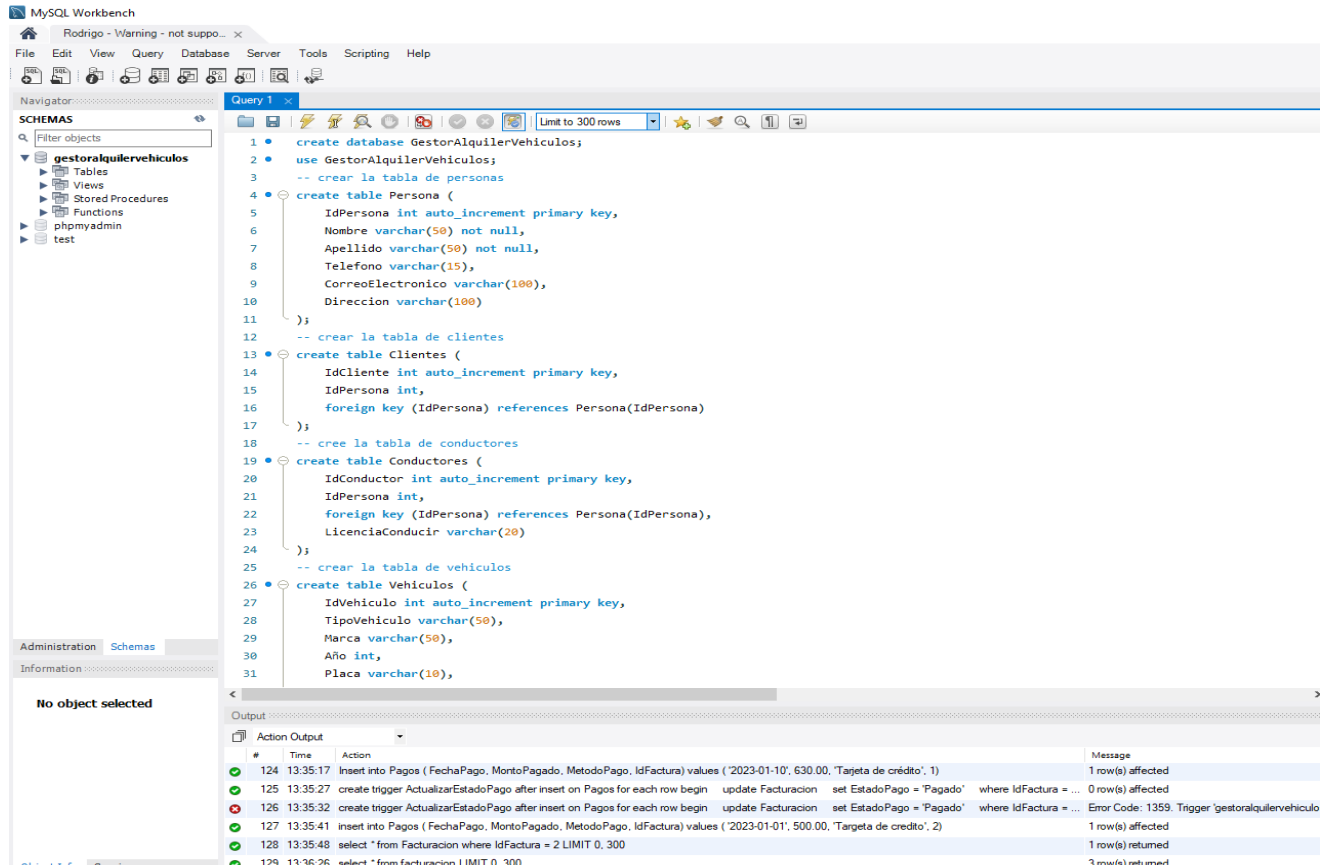
En Este proyecto utilice visual estudio como lenguaje de programación conectando mi base de datos de MySQL.

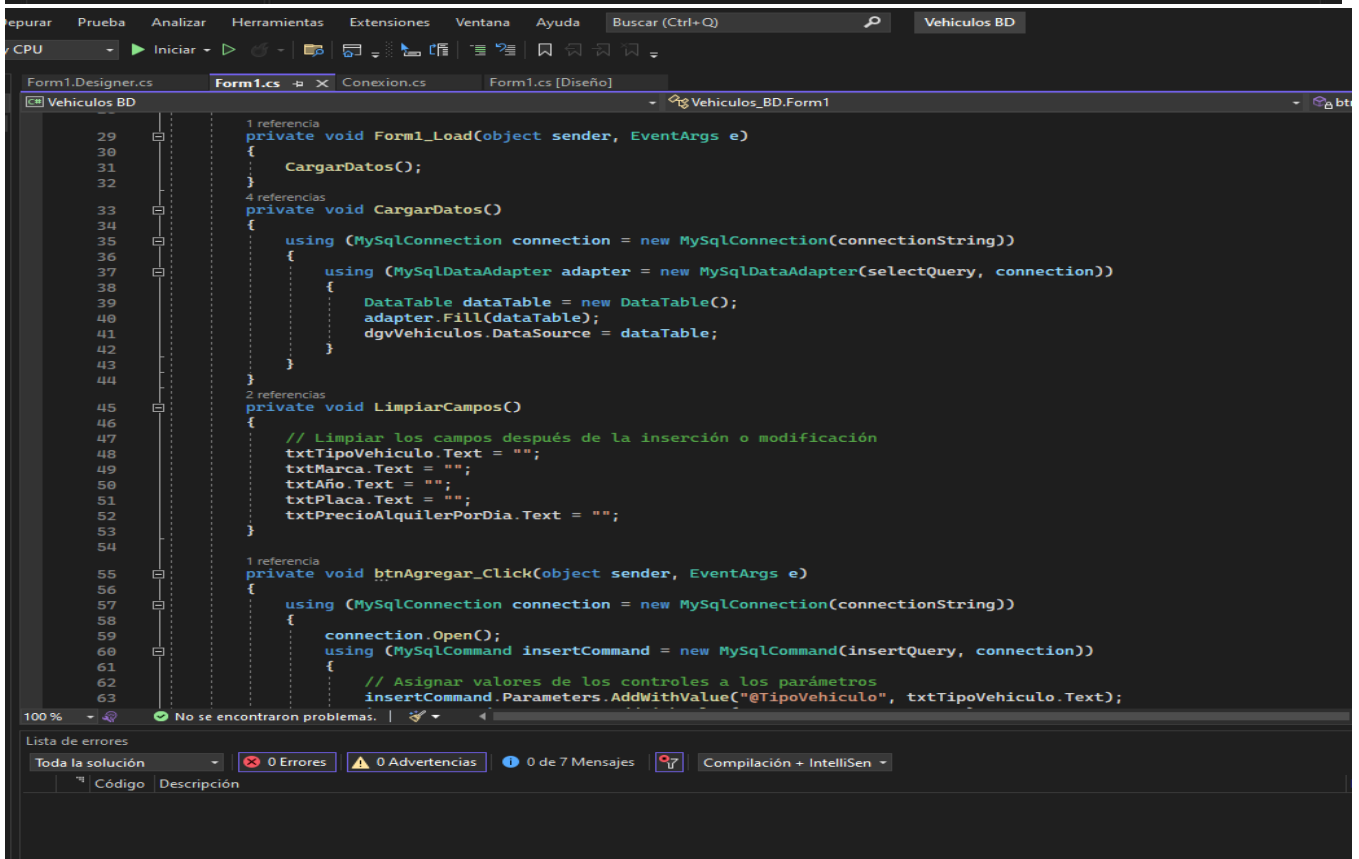
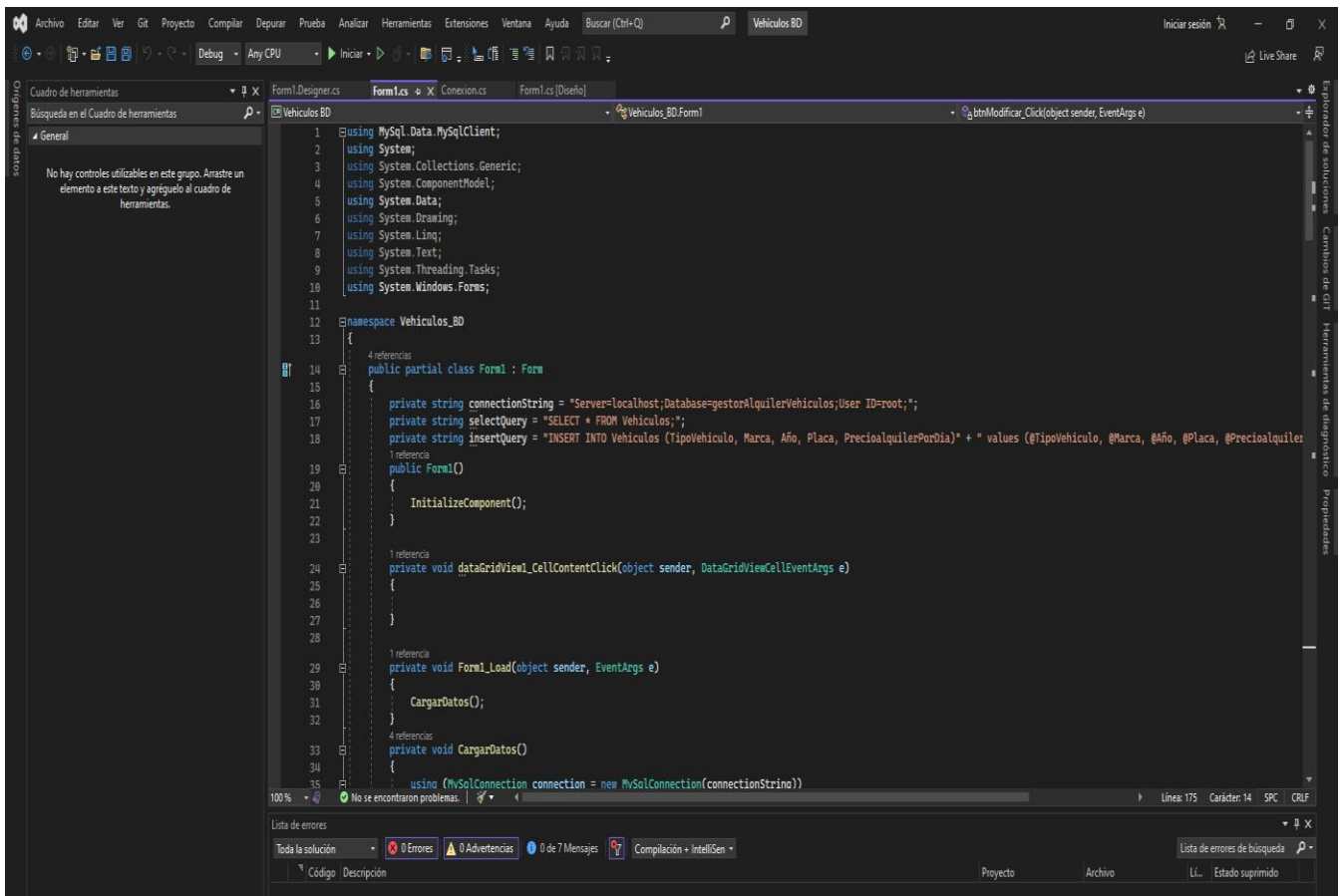
## 5.2 Hardware



## 6.Anexo









```
purar Prueba Analizar Herramientas Extensiones Ventana Ayuda Buscar (Ctrl+Q) Vehiculos BD Iniciar sesión
CPU Iniciar
Form1.Designer.cs Form1.cs Conexion.cs Form1.cs [Diseño]
Vehiculos BD
100 // Recargar los datos en el DataGridView
101 CargarDatos();
102 }
103 else
104 {
105     MessageBox.Show("Selecciona un vehiculo para eliminar.");
106 }
107 }
108
109 1 referencia
110 private void btnModificar_Click(object sender, EventArgs e)
111 {
112     if (dgvVehiculos.SelectedRows.Count > 0)
113     {
114         try
115         {
116             // Obtener el ID del vehiculo seleccionado
117             int idVehiculo = Convert.ToInt32(dgvVehiculos.SelectedRows[0].Cells["IdVehiculo"].Value);
118             // Construir la consulta de actualización
119             string updateQuery = "UPDATE Vehiculos SET TipoVehiculo = @TipoVehiculo, Marca = @Marca, Año = @Año, Placa = @Placa, PrecioalquilerPorDia = @PrecioalquilerPorDia";
120             // Abrir la conexión y ejecutar la consulta de actualización
121             using (MySQLConnection connection = new MySQLConnection(connectionString))
122             {
123                 connection.Open();
124                 using (MySQLCommand updateCommand = new MySQLCommand(updateQuery, connection))
125                 {
126                     // Asignar valores de los controles a los parámetros
127                     updateCommand.Parameters.AddWithValue("@IdVehiculo", idVehiculo);
128                     updateCommand.Parameters.AddWithValue("@TipoVehiculo", txtTipoVehiculo.Text);
129                     updateCommand.Parameters.AddWithValue("@Marca", txtMarca.Text);
130                     // Verificar si la cadena es convertible a un entero para el año
131                     if (int.TryParse(txtAño.Text, out int año))
132                     {
133                         updateCommand.Parameters.AddWithValue("@Año", año);
134                     }
135                 }
136             }
137         }
138         else
139         {
140             // Manejar el error de conversión
141             MessageBox.Show("El año no es un número válido.");
142         }
143     }
144 }
145
146 100% No se encontraron problemas. Línea: 175 Carácter: 14 SPC CRLF
Lista de errores
0 Errores 0 Advertencias 0 de 7 Mensajes Compilación + IntelliSense
Lista de errores de búsqueda
```

```
Prueba Analizar Herramientas Extensiones Ventana Ayuda Buscar (Ctrl+Q) Vehiculos BD Iniciar sesión
Form1.Designer.cs Form1.cs Conexion.cs Form1.cs [Diseño]
Vehiculos BD
1 using MySQL.Data.MySqlClient;
2 using System;
3 using System.Collections.Generic;
4 using System.Data.SqlClient;
5 using System.Linq;
6 using System.Text;
7 using System.Threading.Tasks;
8
9 namespace Basededatos_GestorAlquiler
10 {
11     1 referencia
12     internal class Conexion
13     {
14         private MySqlConnection conexion;
15         private string server = "localhost";
16         private string database = "gestorAlquilerVehiculos";
17         private string user = "root";
18         private string password = "";
19         private string cadenaConexion;
20
21         0 referencias
22         public Conexion()
23         {
24             cadenaConexion = "Database=" + database +
25                 ";Server=" + server +
26                 ";User Id=" + user +
27                 ";Password=" + password;
28         }
29
30         0 referencias
31         public MySqlConnection getConexion()
32         {
33             if (conexion == null)
34             {
35                 conexion = new MySqlConnection(cadenaConexion);
36                 conexion.Open();
37             }
38             return conexion;
39         }
40     }
41 }
% No se encontraron problemas. Línea: 25 Carácter: 14 SPC CRLF
Lista de errores...
0 Errores 0 Advertencias 0 de 13 Mensajes Compilación + IntelliSense
Lista de errores de búsqueda
Código Descripción Proyecto Archivo Línea Estado
```