



# INFORME DE PROYECTO

## HITO 5

**INGENIERIA EN SISTEMAS**

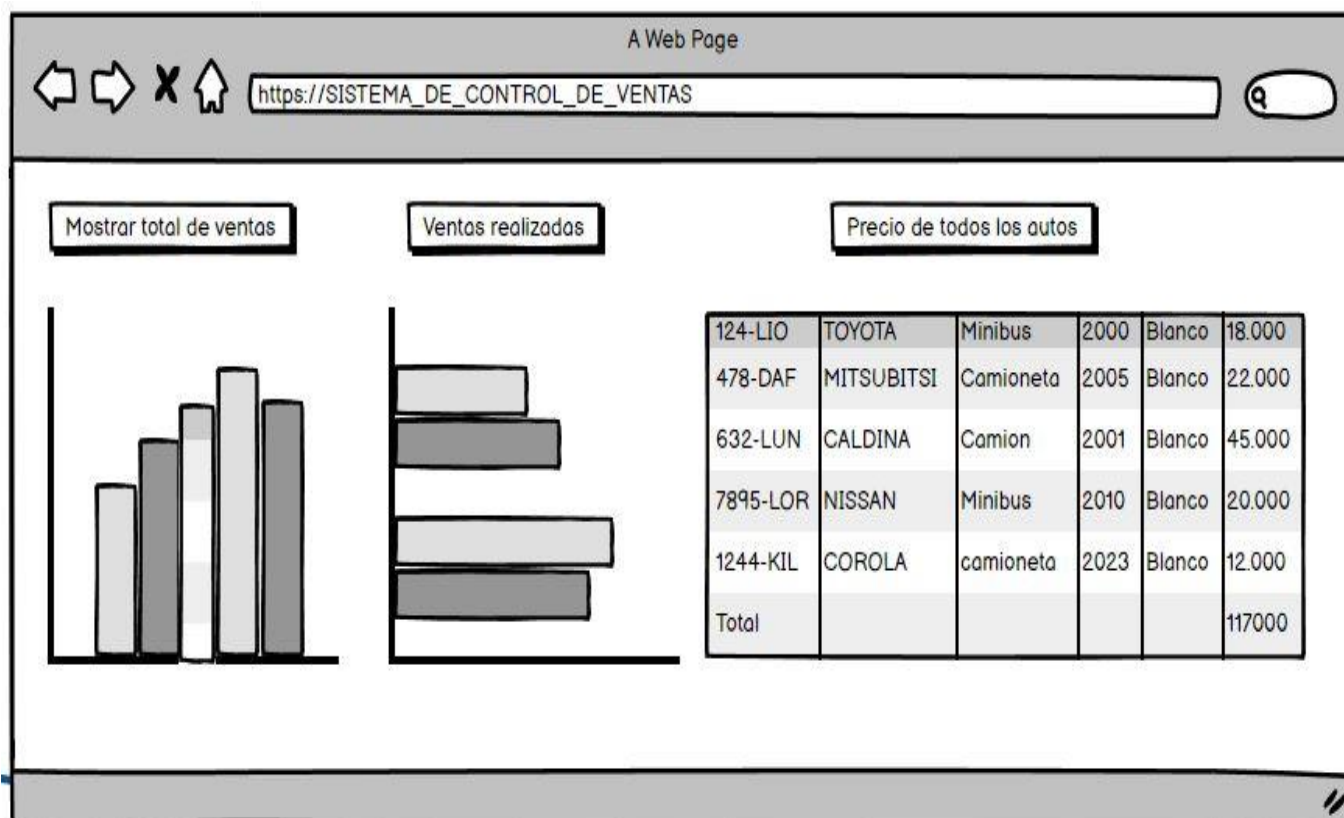
**DOCENTE:** WILLIAM  
BARRA

**INTEGRANTES:** HEBER QUISPE  
CRISTHIAN TANTANI

**GESTION:** 2023

## 1. INTRODUCCION :

En cuanto al contexto del proyecto, se trata de una aplicación de gestión de ventas de vehículos que permite registrar información sobre clientes, vendedores, autos, ventas, financiamientos, pagos, mecánicos y mantenimientos. Esta aplicación puede ser utilizada por una empresa que se dedique a la venta de vehículos para llevar un registro detallado de sus operaciones y tomar decisiones informadas en base a los datos almacenados en la base de datos.





## 2. ANALISIS Y DISEÑO DE BASE DE DATOS: SISTEMA DE CONTROL DE VENTAS

### 3. ENTIDADES TABLA DE SISTEMAS

**CLIENTE** – Almacena los datos del cliente loqueado al sistema.

**VENDEDORES** – Almacena los datos del personal a cargo de las ventas.

**AUTOS** – Son el producto y los modelos que están a la venta.

**VENTAS** – Se registra las ventas realizadas por los vendedores.

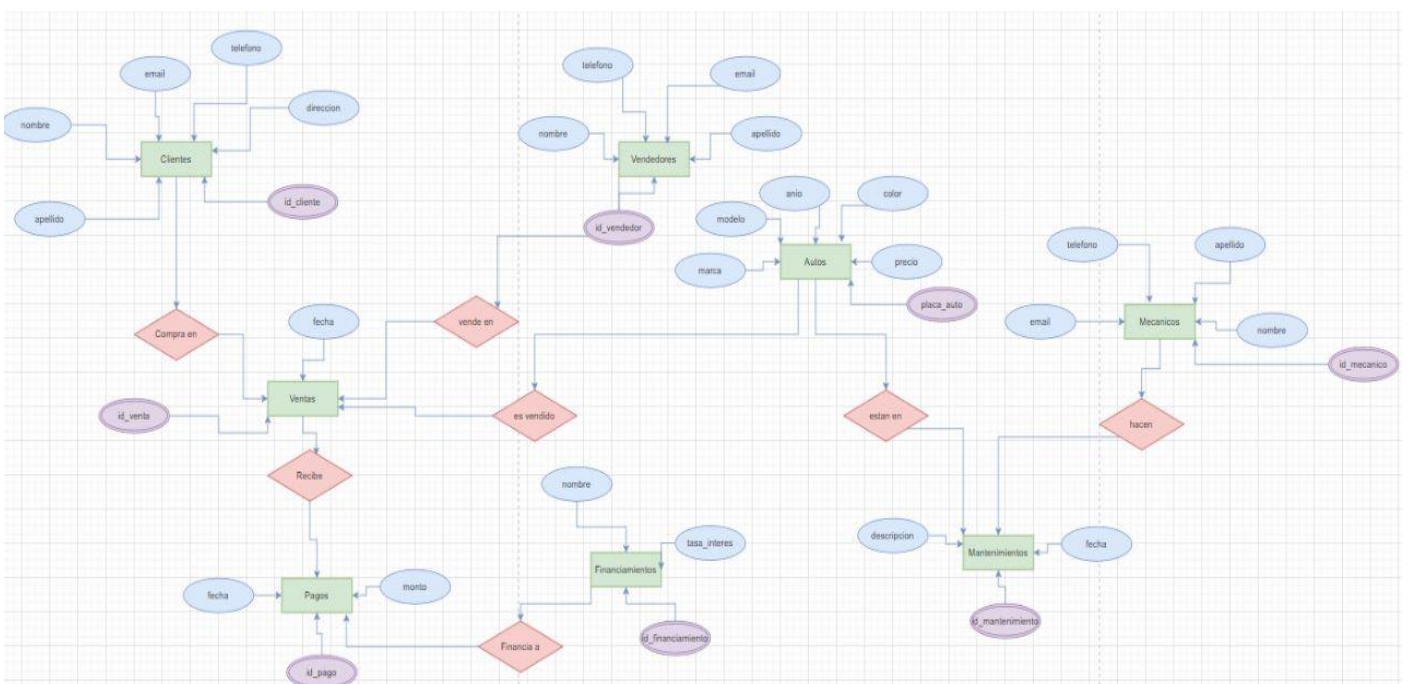
**FINANCIAMIENTOS** – Para asegurar de donde proviene el dinero e tener una garantía.

**PAGOS** – Forma que se paga el auto al contado o a crédito

**MECANICOS** - Personal de trabajo que garantiza cualquier solución al auto vendido.

**MANTENIMIENTOS** – Poder realizar revisiones a los automóviles y garantizar su uso.

### DISEÑO DE LA BASE DE DATOS:





# ARCHIVO SQL

```
CREATE DATABASE SISTEMA_DE_CONTROL_DE_VENTAS;
```

```
USE SISTEMA_DE_CONTROL_DE_VENTAS;
```

```
CREATE TABLE Clientes
```

```
(  
    id_cliente INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    apellido VARCHAR(50),  
    telefono VARCHAR(20),  
    email VARCHAR(50),  
    direccion VARCHAR(100)  
);
```

```
INSERT INTO Clientes VALUES (123, 'Juan', 'Perez', '78944848', 'juan@gmail.com', 'Av.Juan_Pablo');
```

```
INSERT INTO Clientes VALUES (1234, 'Luis', 'Mamamni', '85944848', 'luis@gmail.com', 'Av.La_paz');
```

```
INSERT INTO Clientes VALUES (12345, 'Pepe', 'Alanoca', '6984848', 'pepe@gmail.com', 'Av.Bolivia');
```

```
INSERT INTO Clientes VALUES (123456, 'Jose', 'Quisbert', '785944848', 'jose@gmail.com', 'Av.Perez');
```

```
INSERT INTO Clientes VALUES (1234567, 'Daniel', 'Mayta', '12344848', 'dani@gmail.com', 'Av.Litoral');
```

```
CREATE TABLE Vendedores
```

```
(  
    id_vendedor INT PRIMARY KEY,  
    nombre VARCHAR(50),  
    apellido VARCHAR(50),  
    telefono VARCHAR(20),  
    email VARCHAR(50)
```

);

```
INSERT INTO Vendedores VALUES (1, 'Alex', 'Machicado', '4547874', 'alex@gmail.com');
```

```
INSERT INTO Vendedores VALUES (2, 'Alberto', 'Barra', '54747874', 'albert@gmail.com');
```

```
CREATE TABLE Autos
```

```
(
```

```
placa_auto VARCHAR(50) PRIMARY KEY,
```

```
marca VARCHAR(50),
```

```
modelo VARCHAR(50),
```

```
anio INT,
```

```
color VARCHAR(20),
```

```
precio DECIMAL(10, 2)
```

```
);
```

```
INSERT INTO Autos VALUES ('124-L10', 'TOYOTA', 'Minibus', 2000, 'Blanco', '18.000');
```

```
INSERT INTO Autos VALUES ('478-DAF', 'MITSUBITS', 'Camioneta', 2005, 'Blanco', '22.000');
```

```
INSERT INTO Autos VALUES ('632-LUN', 'CALDINA', 'Camion', 2001, 'Blanco', '45.000');
```

```
INSERT INTO Autos VALUES ('7895-LOR', 'NISSAN', 'Minibus', 2010, 'Blanco', '20.000');
```

```
INSERT INTO Autos VALUES ('1244-KIL', 'CORDA', 'camioneta', 2023, 'Blanco', '12.000');
```

```
CREATE TABLE Ventas
```

```
(
```

```
id_venta INT PRIMARY KEY,
```

```
fecha DATE,
```

```
id_cliente INT,
```

```
id_vendedor INT,
```

```
placa_auto VARCHAR(50),
```

```
FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente),
```

```
FOREIGN KEY (id_vendedor) REFERENCES Vendedores(id_vendedor),
```

```
FOREIGN KEY (placa_auto) REFERENCES Autos(placa_auto)
```

```
);
```

```
CREATE TABLE Financiamientos
```

```
(
```

```
id_financiamiento INT PRIMARY KEY,
```

```
nombre VARCHAR(50),
```

```
tasa_interes DECIMAL(5, 2)
```

```
);
```

```
CREATE TABLE Pagos
```

```
(
```

```
id_pago INT PRIMARY KEY,
```

```
fecha DATE,
```

```
monto DECIMAL(10, 2),
```

```
id_venta INT,
```

```
id_financiamiento INT,
```

```
FOREIGN KEY (id_venta) REFERENCES Ventas(id_venta),
```

```
FOREIGN KEY (id_financiamiento) REFERENCES Financiamientos(id_financiamiento)
```

```
);
```

```
CREATE TABLE Mecanicos
```

```
(
```

```
id_mecanico INT PRIMARY KEY,
```

```
nombre VARCHAR(50),
```

```
apellido VARCHAR(50),
```

```
telefono VARCHAR(20),
```

```
email VARCHAR(50)
```

```
);
```

```
INSERT INTO Mecanicos VALUES (1, 'Dante', 'Alanoca', '4785777', 'alan@gmail.com');
```

```
INSERT INTO Mecanicos VALUES (2, 'Robert', 'Tinta', '7885777', 'tinta@gmail.com');
```

```
CREATE TABLE Mantenimientos
```

```
(
```

```
id_mantenimiento INT PRIMARY KEY,
```

```
fecha DATE,
```

```
descripcion VARCHAR(200),
```

```
placa_auto VARCHAR(50),
```

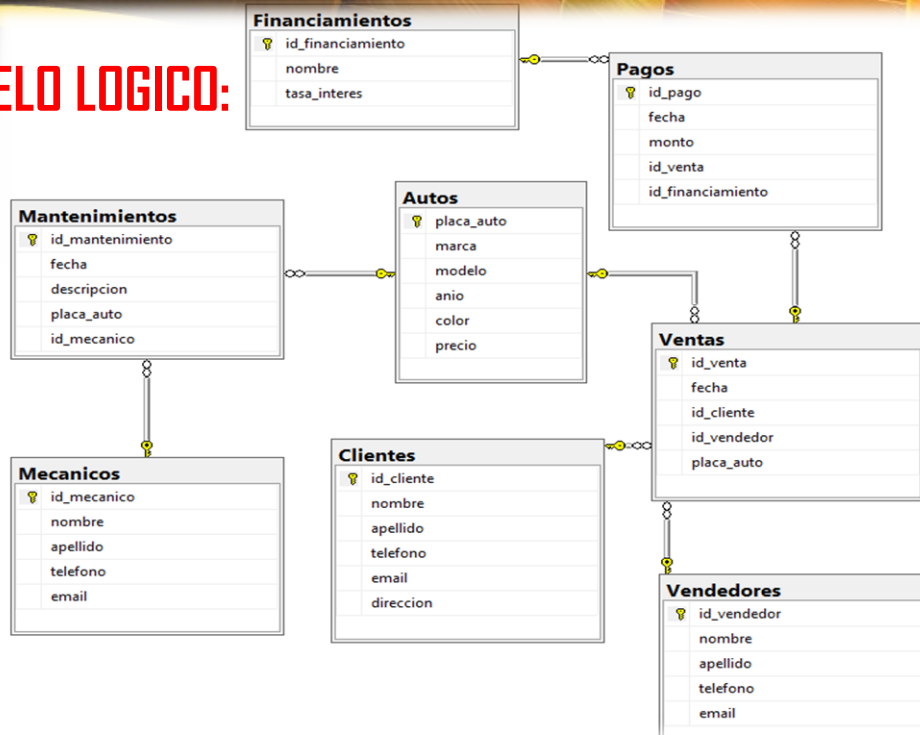
```
id_mecanico INT,
```

```
FOREIGN KEY (placa_auto) REFERENCES Autos(placa_auto),
```

```
FOREIGN KEY (id_mecanico) REFERENCES Mecanicos(id_mecanico)
```

```
);
```

## MODELO LOGICO:



## NOTAS: MANEJO DE JOINS Y CONSULTAS


--Mostrar el nombre del cliente, el nombre del vendedor y la marca y modelo del auto vendido por cada venta realizada:

```
SELECT Clientes.nombre, Vendedores.nombre, Autos.marca, Autos.modelo
FROM Ventas
JOIN Clientes ON Ventas.id_cliente = Clientes.id_cliente
JOIN Vendedores ON Ventas.id_vendedor = Vendedores.id_vendedor
JOIN Autos ON Ventas.placa_auto = Autos.placa_auto;
```

--Mostrar el nombre del cliente la fecha de la venta y el monto pagado para cada venta realizada con financiamiento:

```
SELECT Clientes.nombre, Ventas.fecha, Pagos.monto
FROM Ventas
JOIN Clientes ON Ventas.id_cliente = Clientes.id_cliente
JOIN Pagos ON Ventas.id_venta = Pagos.id_venta
JOIN Financiamientos ON Pagos.id_financiamiento =
Financiamientos.id_financiamiento;
```





--Mostrar el nombre del mecánico, la fecha del mantenimiento y la marca y modelo del auto por cada mantenimiento que se iso:

```
SELECT Mecanicos.nombre, Mantenimientos.fecha, Autos.marca, Autos.modelo
FROM Mantenimientos
JOIN Mecanicos ON Mantenimientos.id_mecanico = Mecanicos.id_mecanico
JOIN Autos ON Mantenimientos.placa_auto = Autos.placa_auto;
```

--Mostrar el nombre del cliente y la cantidad total de dinero que ha gastado en compras de autos:

```
SELECT Clientes.nombre, SUM(Autos.precio)
FROM Ventas
JOIN Clientes ON Ventas.id_cliente = Clientes.id_cliente
JOIN Autos ON Ventas.placa_auto = Autos.placa_auto
```

--Mostrar el nombre del vendedor y la cantidad total de dinero que ha ganado en comisiones por ventas realizadas:

```
SELECT Vendedores.nombre, SUM(Autos.precio * 0.1)
FROM Ventas
JOIN Vendedores ON Ventas.id_vendedor = Vendedores.id_vendedor
JOIN Autos ON Ventas.placa_auto = Autos.placa_auto
GROUP BY Vendedores.nombre;
```

## FUNCIONES DE AGREGACION

Suma de los precios de todos los autos en la tabla "Autos"

Contar el número de ventas realizadas en la tabla "Ventas"

```
SELECT COUNT(*) AS "Número de ventas" FROM Ventas;
```

Obtener el promedio de precios de los autos en la tabla "Autos"

Obtener el año más reciente de los autos en la tabla "Autos"

```
SELECT MAX(anio) AS "Año más reciente" FROM Autos;
```

Obtener el color más oscuro de los autos en la tabla "Autos"

```
SELECT MIN(color) AS "Color más oscuro" FROM Autos;
```



## FUNCIONES DE AGREGACION II

--obtener el total de ventas realizadas por cada vendedor en la tabla "Ventas"

```
SELECT Vendedores.nombre, SUM(Pagos.monto) AS "Total de ventas"
FROM Ventas
JOIN Vendedores ON Ventas.id_vendedor = Vendedores.id_vendedor
JOIN Pagos ON Ventas.id_venta = Pagos.id_venta
```

--obtener el color más oscuro de los autos vendidos por cada vendedor en la tabla "Ventas"

```
SELECT Vendedores.nombre, MIN(Autos.color) AS "Color más oscuro"
FROM Ventas
JOIN Vendedores ON Ventas.id_vendedor = Vendedores.id_vendedor
JOIN Autos ON Ventas.placa_auto = Autos.placa_auto
```

--obtener el año más reciente de los autos vendidos por cada vendedor en la tabla "Ventas"

```
SELECT Vendedores.nombre, MAX(Autos.anio) AS "Año más reciente"
FROM Ventas
JOIN Vendedores ON Ventas.id_vendedor = Vendedores.id_vendedor
JOIN Autos ON Ventas.placa_auto = Autos.placa_auto
```

--obtener el promedio de precios de los autos vendidos por cada vendedor en la tabla "Ventas"

```
SELECT Vendedores.nombre, AVG(Autos.precio) AS "Precio promedio"
FROM Ventas
JOIN Vendedores ON Ventas.id_vendedor = Vendedores.id_vendedor
JOIN Autos ON Ventas.placa_auto = Autos.placa_auto
```

--Mostrar el nombre del cliente y la cantidad total de dinero que ha gastado en compras de autos:

```
SELECT Clientes.nombre, SUM(Autos.precio)
FROM Ventas
JOIN Clientes ON Ventas.id_cliente = Clientes.id_cliente
JOIN Autos ON Ventas.placa_auto = Autos.placa_auto
```

## FUNCIONES UDF

**obtener el promedio de precios de los autos en la tabla "Autos"**

```
CREATE FUNCTION dbo.TotalVentas ()
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @total DECIMAL(10, 2);
    SELECT @total = SUM(precio) FROM Autos;
    RETURN @total;
END;
```

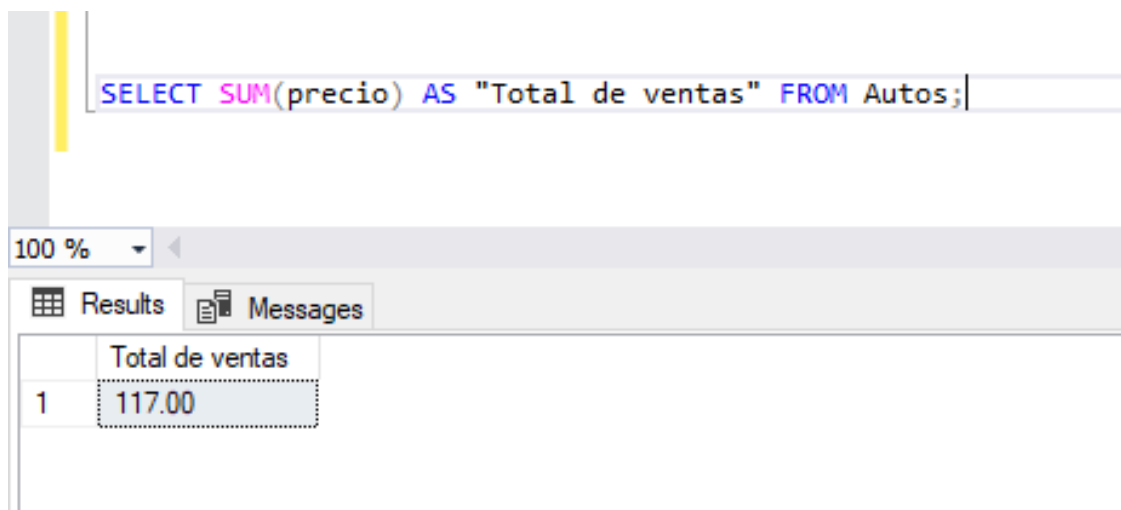
### contar el número de ventas realizadas en la tabla "Ventas"

```
CREATE FUNCTION dbo.NumVentas ()
RETURNS INT
AS
BEGIN
    DECLARE @num_ventas INT;
    SELECT @num_ventas = COUNT(*) FROM Ventas;
    RETURN @num_ventas;
END;
```

### obtener la suma de los precios de todos los autos en la tabla "Autos"

```
CREATE FUNCTION dbo.TotalVentas ()
RETURNS DECIMAL(10, 2)
AS
BEGIN
    DECLARE @total DECIMAL(10, 2);
    SELECT @total = SUM(precio) FROM Autos;
    RETURN @total;
END;
```

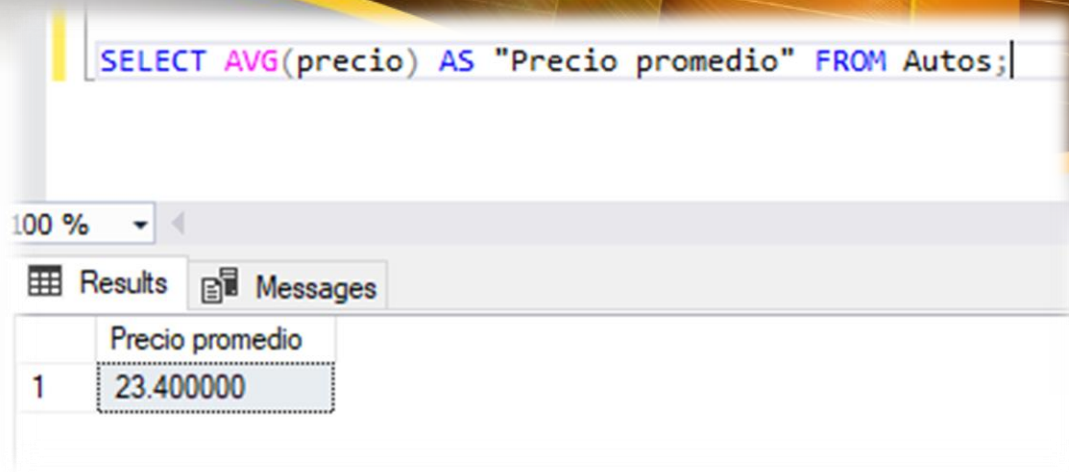
## USABILIDAD



The screenshot shows a SQL query window with the following text: `SELECT SUM(precio) AS "Total de ventas" FROM Autos;`

Below the query window, the 'Results' tab is active, displaying a single row of data:

	Total de ventas
1	117.00



## CONCLUSIONES

El objetivo de este proyecto era poder crear una base de datos que pueda registrar ventas de vehículos echas para una empresa

La base de datos SISTEMA DE CONTROL DE VENTAS funciona como una base de datos y tiene las 8 tablas que se requirió todas relacionadas al base de datos, se pueden realizar consultas hacer funciones es funcional

Nos hubiera gustado poder relacionarlo con visual estudio para hacer una interfaz gráfica con botones imágenes etc. Por esa parte nos gustaría poder mejorarlo