**UNIDAD 9**

**Proyecto de BBDD Relacionales**



David Vazquez Novoa .

**ÍNDICE:**

[1. Recogida de Requisitos](#_hwvmakxioey)

[1.1.Interactuar con el bot-cliente para obtener los requisitos.](#_4jq6llpipxef)

[1.2.Documentar la conversación completa en un archivo](#_km0mdioom1sd)

[1.3.Sintetizar los requisitos funcionales y no funcionales.](#_fci1w2pev92)

[Requisitos funcionales:](#_6cwrm6wm3rd5)

[Requisitos No Funcionales](#_aoyy4hhnt1bw)

[1.4.Tablas que me hacen falta para la base de datos:](#_9kyd7xthg78c)

# 

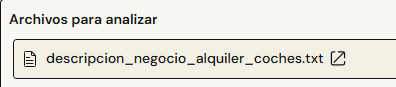
# 1. Recogida de Requisitos

### 1.1.Interactuar con el bot-cliente para obtener los requisitos.

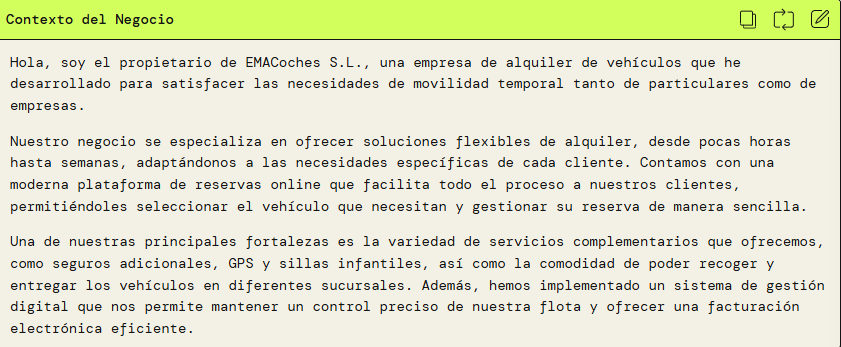
Lo primero que hago es elegir el tipo de negocio en el que se base el proyecto, elegí Servicio de alquiles de coches.

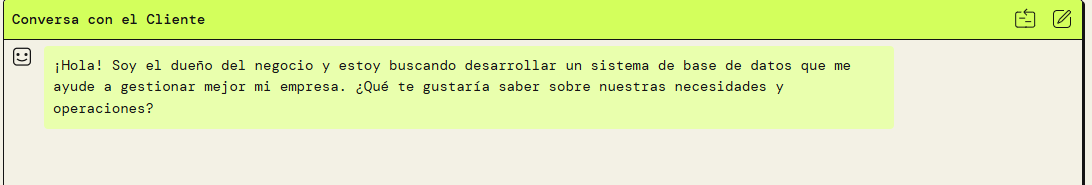
### 1.2.Documentar la conversación completa en un archivo

Despues creo un documento y hago la descripción del negocio y la subo a ‘Archivos para analizar’,



El boot después de entregarle ese archivo crea un texto :





Ahora le contesto, lo que que me gustaría saber:

¿Qué datos desea guardar de los clientes?

¿Qué información necesita almacenar de los vehículos?

¿Cómo funciona el proceso de reserva y devolución?

¿Qué métodos de pago acepta y cómo los gestiona?

¿Quiere poder consultar el estado de los vehículos (disponibles, en alquiler, etc.)?

### 1.3.Sintetizar los requisitos funcionales y no funcionales.

#### Requisitos funcionales:

Clientes

* Registrar, modificar y eliminar información de clientes.
* Validar el número de carnet de conducir y su fecha de caducidad.
* Guardar historial de alquileres por cliente.
* Registrar incidencias u observaciones asociadas a cada cliente.

Vehículos

* Registrar, modificar y eliminar vehículos.
* Consultar características y estado de los vehículos.
* Controlar disponibilidad (Disponible, Reservado, En alquiler, etc.).
* Gestionar el historial de mantenimiento y documentación.

Reservas y alquileres

* Permitir registrar una reserva con fechas, horarios, sucursales y servicios extra.
* Controlar la recogida y devolución de vehículos (con kilometraje y nivel de combustible).
* Registrar incidencias durante el alquiler.

Pagos y facturación

* Gestionar diferentes métodos de pago (tarjeta, PayPal, efectivo…).
* Generar y asociar facturas a las reservas.
* Controlar depósitos de seguridad.

Otros

* Consultar ubicación y próxima revisión de cada coche.
* Generar informes del historial de alquileres por cliente, vehículo o periodo.

#### Requisitos No Funcionales

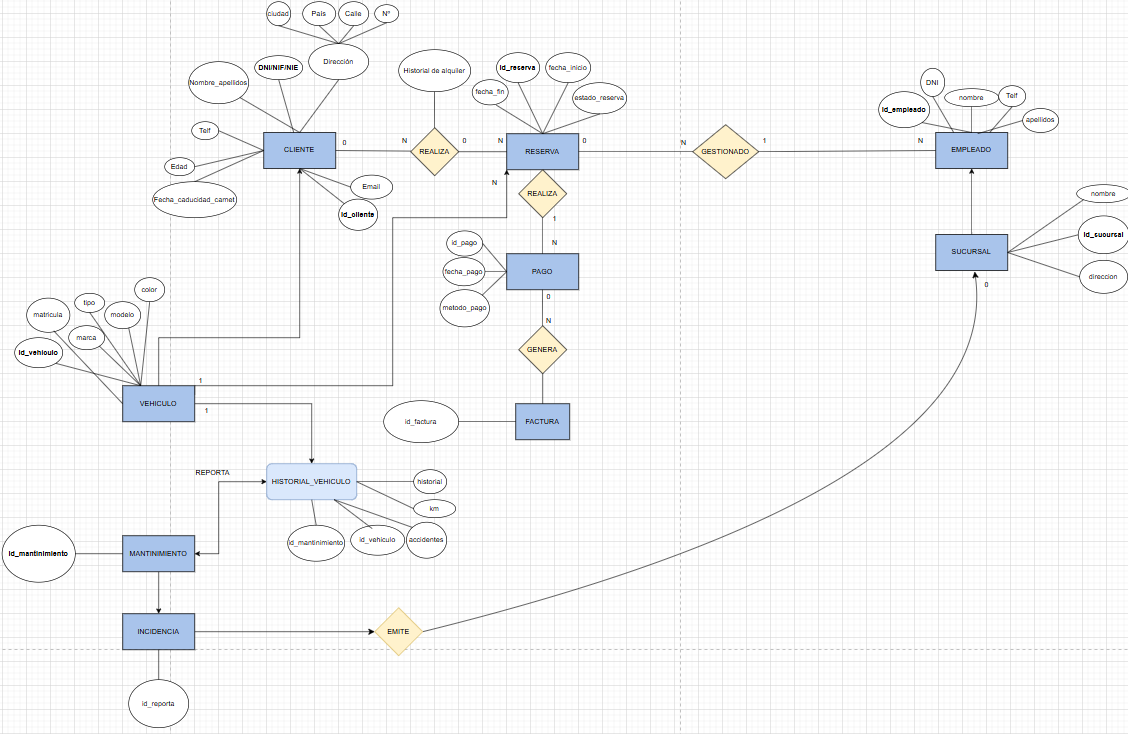
Son las características de calidad del sistema.

* El sistema debe ser accesible desde diferentes dispositivos (móvil, PC).
* Debe garantizar la integridad de los datos (uso de claves, restricciones).
* El acceso a los datos debe ser rápido y eficiente.
* El sistema debe permitir copias de seguridad periódicas.
* Debe usarse autenticación para que solo personal autorizado gestione datos.
* Debe permitir auditoría de accesos y operaciones críticas.

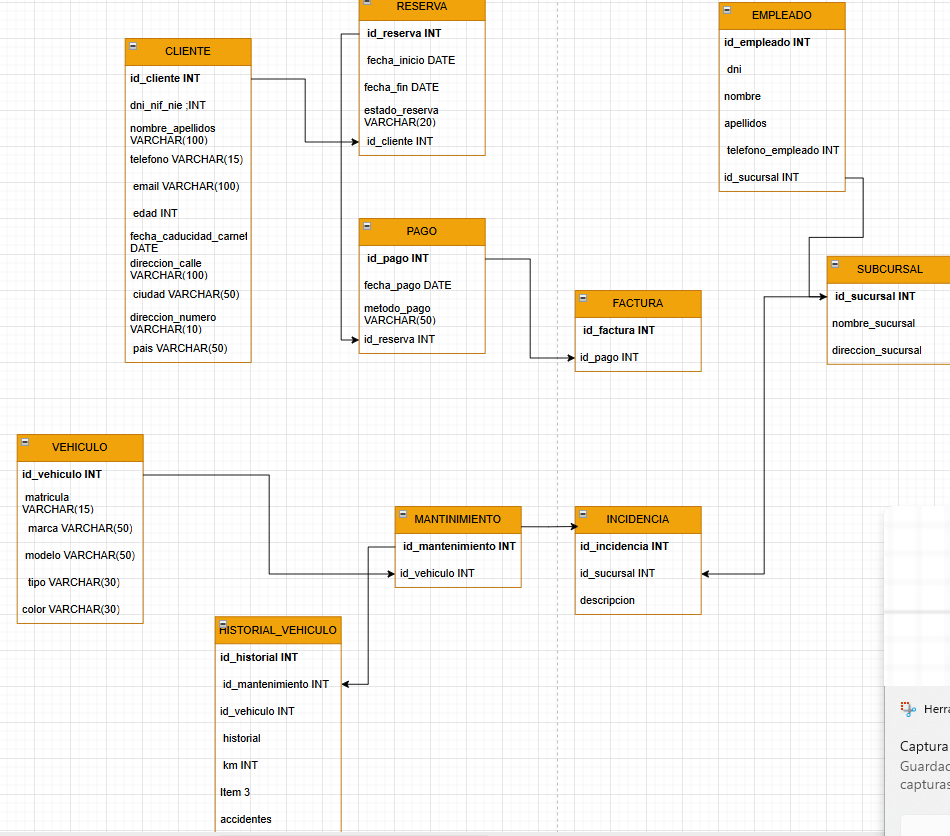
### 1.4.Tablas que me hacen falta para la base de datos:

|  |  |  |
| --- | --- | --- |
| **Nº** | **Tabla** | **Descripción** |
| 1 | **clientes** | Datos personales, contacto, carnet |
| 2 | **vehículos** | Datos del coche, estado, características |
| 3 | **reservas** | Info de la reserva (fechas, cliente, vehículo) |
| 4 | **empleados** | Quien atiende las reservas, mantenimiento, etc. |
| 5 | **sucursales** | Lugares de recogida/devolución |
| 6 | **pagos** | Datos de cada pago y método usado |
| 7 | **facturas** | Registro de facturas asociadas a pagos |
| 8 | **mantenimientos** | Historial de mantenimiento de vehículos |
| 9 | **incidencias** | Incidencias durante el alquiler o del cliente |
| 10 | **servicios\_adicionales** | GPS, silla bebé, seguro extra... |
| 11 | **reserva\_servicio** | Tabla intermedia para relacionar reservas con servicios adicionales (N:N) |
|  |  |  |

# 2. Modelado Conceptual



# 3. Modelo Relacional



[imagen decorar portada]

***url Github***

**Nome  Alumno/a:** [**David Vázquez Nóvoa**](mailto:david.vazquez.novoa@iessanmamede.com)

***Nome completo e apelidos***

**Curso:  *1º DAM*                Materia:** ***Bases de Datos – Proyecto Final 24/25***

Contido

[1. Introducción 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.3t4eedwf8up5)

[2. Descripción del Problema / Requisitos 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.u5vugrsrso2)

[3. Modelo Conceptual 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.jeh84w2l7f6i)

[4. Modelo Relacional 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.5mgux8hjrm5l)

[5. Proceso de Normalización 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.edlr91aewso2)

[6. Script de Creación de la Base de Datos 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.3wt94ssmro3z)

[7. Carga de Datos Inicial 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.n6v5dp2m1e0h)

[8. Funciones y Procedimientos Almacenados 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.rj01hssdayfr)

[9. Triggers 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.8p8n2bxl6p1g)

[10. Consultas SQL 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.3gtubbhkikfc)

[11. Casos de Prueba y Simulación 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.ngj0sq3fzq6v)

[12. Resultados y Verificación 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.qqs1e9o0zvf)

[13. Capturas de Pantalla (opcional) 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.f8slzgc27eyo)

[14. Conclusiones y Mejoras Futuras 2](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.mtlrlb3pp2le)

[15. Enlace al Repositorio en GitHub 3](https://docs.google.com/document/d/1oDZZQ7AglJN_RbAqzlBnIhO2tt2a2g6cmGHghw1UDUo/edit#heading=h.q4uk03tbmsep)

# Introducción

Describe aquí...

# Descripción del Problema / Requisitos

#### Requisitos funcionales:

Clientes

●     Registrar, modificar y eliminar información de clientes.

●     Validar el número de carnet de conducir y su fecha de caducidad.

●     Guardar historial de alquileres por cliente.

●     Registrar incidencias u observaciones asociadas a cada cliente.

Vehículos

●     Registrar, modificar y eliminar vehículos.

●     Consultar características y estado de los vehículos.

●     Controlar disponibilidad (Disponible, Reservado, En alquiler, etc.).

●     Gestionar el historial de mantenimiento y documentación.

Reservas y alquileres

●     Permitir registrar una reserva con fechas, horarios, sucursales y servicios extra.

●     Controlar la recogida y devolución de vehículos (con kilometraje y nivel de combustible).

●     Registrar incidencias durante el alquiler.

Pagos y facturación

●     Gestionar diferentes métodos de pago (tarjeta, PayPal, efectivo…).

●     Generar y asociar facturas a las reservas.

●     Controlar depósitos de seguridad.

Otros

●     Consultar ubicación y próxima revisión de cada coche.

●     Generar informes del historial de alquileres por cliente, vehículo o periodo.

#### Requisitos No Funcionales

Son las características de calidad del sistema.

●     El sistema debe ser accesible desde diferentes dispositivos (móvil, PC).

●     Debe garantizar la integridad de los datos (uso de claves, restricciones).

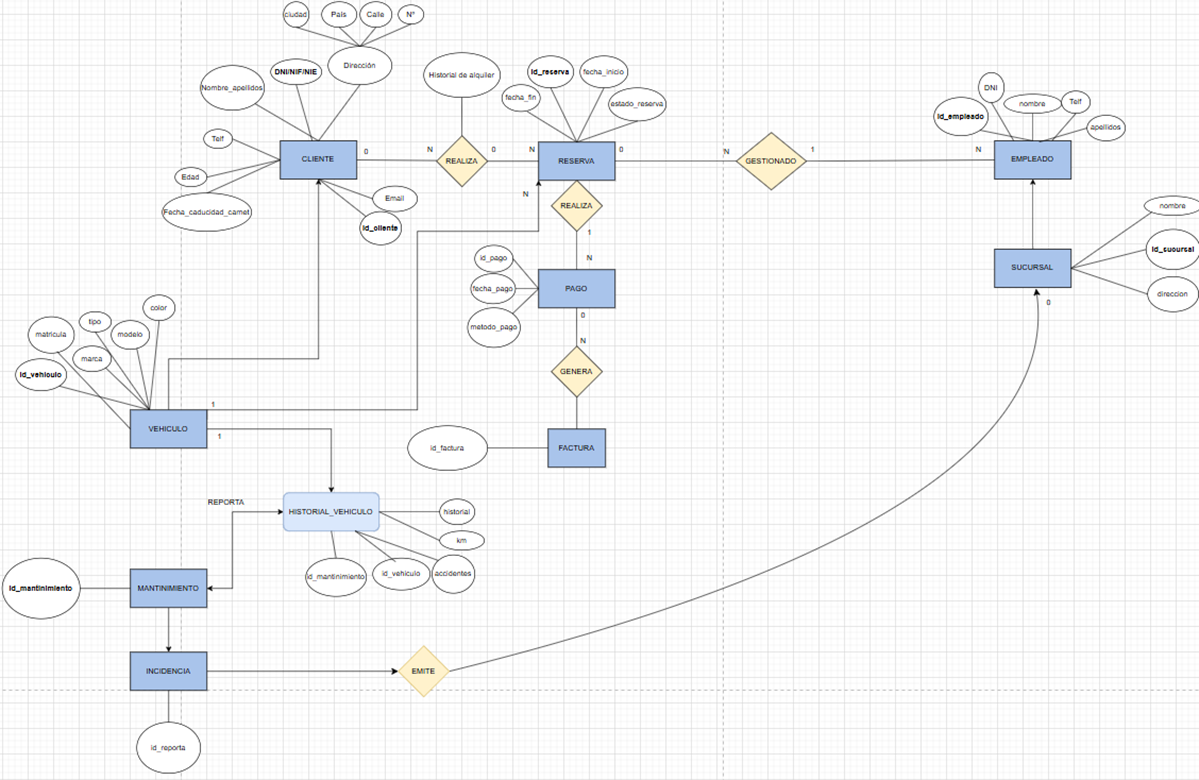
●     El acceso a los datos debe ser rápido y eficiente.

●     El sistema debe permitir copias de seguridad periódicas.

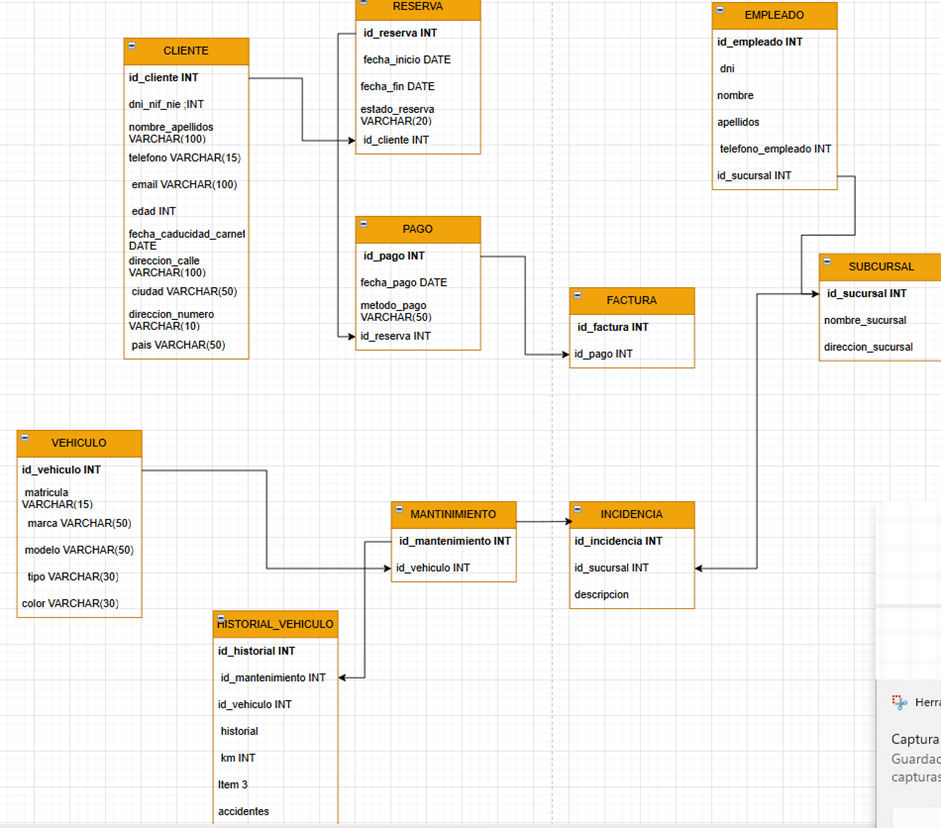
●     Debe usarse autenticación para que solo personal autorizado gestione datos.

●     Debe permitir auditoría de accesos y operaciones críticas.

# Modelo Conceptual



# Modelo Relacional



# Proceso de Normalización

**Primera Forma Normal (1FN):**

* Todos los atributos son atómicos.
* No existen grupos repetitivos ni atributos multivaluados.
* Ejemplo: la dirección se descompone en calle, número, ciudad y país.
* Se cumple 1FN.

**Segunda Forma Normal (2FN):**

* Se cumple 1FN.
* No hay dependencias parciales, ya que todas las claves primarias son simples.
* Todos los atributos no clave dependen completamente de la clave primaria.
* Se cumple 2FN.

**Tercera Forma Normal (3FN):**

1. Se cumple 2FN.
2. No hay dependencias transitivas.
3. Ejemplo: en la tabla CLIENTE, ningún atributo no clave depende de otro atributo no clave.
4. Se cumple 3FN.

# Script de Creación de la Base de Datos

CREATE DATABASE extremcar;

USE extremcar;

-- Tabla CLIENTE

CREATE TABLE CLIENTE (

    id\_cliente INT PRIMARY KEY,

    dni\_nif\_nie VARCHAR(20) NOT NULL,

    nombre\_apellidos VARCHAR(100),

    telefono VARCHAR(15),

    email VARCHAR(100),

    edad INT,

    fecha\_caducidad\_carnet DATE,

    direccion\_calle VARCHAR(100),

    direccion\_numero VARCHAR(10),

    ciudad VARCHAR(50),

    pais VARCHAR(50)

);

-- Tabla RESERVA

CREATE TABLE RESERVA (

    id\_reserva INT PRIMARY KEY,

    fecha\_inicio DATE,

    fecha\_fin DATE,

    estado\_reserva VARCHAR(20),

    id\_cliente INT,

    FOREIGN KEY (id\_cliente) REFERENCES CLIENTE(id\_cliente)

);

-- Tabla PAGO

CREATE TABLE PAGO (

    id\_pago INT PRIMARY KEY,

    fecha\_pago DATE,

    metodo\_pago VARCHAR(50),

    id\_reserva INT,

    FOREIGN KEY (id\_reserva) REFERENCES RESERVA(id\_reserva)

);

-- Tabla FACTURA

CREATE TABLE FACTURA (

    id\_factura INT PRIMARY KEY,

    id\_pago INT,

    FOREIGN KEY (id\_pago) REFERENCES PAGO(id\_pago)

);

-- Tabla SUCURSAL

CREATE TABLE SUCURSAL (

    id\_sucursal INT PRIMARY KEY,

    nombre\_sucursal VARCHAR(50),

    direccion\_sucursal VARCHAR(100)

);

-- Tabla EMPLEADO

CREATE TABLE EMPLEADO (

    id\_empleado INT PRIMARY KEY,

    dni VARCHAR(20),

    nombre VARCHAR(50),

    apellidos VARCHAR(50),

    telefono\_empleado INT,

    id\_sucursal INT,

    FOREIGN KEY (id\_sucursal) REFERENCES SUCURSAL(id\_sucursal)

);

-- Tabla VEHICULO

CREATE TABLE VEHICULO (

    id\_vehiculo INT PRIMARY KEY,

    matricula VARCHAR(15) UNIQUE,

    marca VARCHAR(50),

    modelo VARCHAR(50),

    tipo VARCHAR(30),

    color VARCHAR(30)

);

-- Tabla MANTENIMIENTO

CREATE TABLE MANTENIMIENTO (

    id\_mantenimiento INT PRIMARY KEY,

    id\_vehiculo INT,

    FOREIGN KEY (id\_vehiculo) REFERENCES VEHICULO(id\_vehiculo)

);

-- Tabla HISTORIAL\_VEHICULO

CREATE TABLE HISTORIAL\_VEHICULO (

    id\_historial INT PRIMARY KEY,

    id\_mantenimiento INT,

    id\_vehiculo INT,

    historial TEXT,

    km INT,

    accidentes TEXT,

    FOREIGN KEY (id\_mantenimiento) REFERENCES MANTENIMIENTO(id\_mantenimiento),

    FOREIGN KEY (id\_vehiculo) REFERENCES VEHICULO(id\_vehiculo)

);

-- Tabla INCIDENCIA

CREATE TABLE INCIDENCIA (

    id\_incidencia INT PRIMARY KEY,

    id\_sucursal INT,

    descripcion TEXT,

    FOREIGN KEY (id\_sucursal) REFERENCES SUCURSAL(id\_sucursal)

);

# Carga de Datos Inicial

INSERT INTO CLIENTE VALUES

(1, '12345678A', 'Juan Pérez', '600000001', 'juanp@example.com', 30, '2026-05-01', 'Calle Sol', '1', 'Madrid', 'España'),

(2, '23456789B', 'María Gómez', '600000002', 'mariag@example.com', 28, '2027-06-15', 'Avenida Luna', '2', 'Barcelona', 'España'),

(3, '34567890C', 'Luis Martínez', '600000003', 'luism@example.com', 35, '2025-07-20', 'Calle Estrella', '3', 'Valencia', 'España'),

(4, '45678901D', 'Ana López', '600000004', 'anal@example.com', 40, '2026-08-25', 'Paseo Marítimo', '4', 'Sevilla', 'España'),

(5, '56789012E', 'Carlos Ruiz', '600000005', 'carlosr@example.com', 50, '2024-09-30', 'Camino Real', '5', 'Bilbao', 'España'),

(6, '67890123F', 'Laura Fernández', '600000006', 'lauraf@example.com', 27, '2027-03-12', 'Calle Norte', '6', 'Zaragoza', 'España'),

(7, '78901234G', 'Pedro Sánchez', '600000007', 'pedros@example.com', 32, '2025-04-18', 'Calle Sur', '7', 'Granada', 'España'),

(8, '89012345H', 'Lucía Torres', '600000008', 'luciat@example.com', 29, '2026-11-05', 'Calle Este', '8', 'Vigo', 'España'),

(9, '90123456I', 'David Morales', '600000009', 'davidm@example.com', 31, '2026-01-09', 'Calle Oeste', '9', 'Oviedo', 'España'),

(10, '01234567J', 'Elena Rivas', '600000010', 'elenar@example.com', 36, '2027-10-22', 'Calle Centro', '10', 'A Coruña', 'España');

INSERT INTO VEHICULO VALUES

(1, '1234ABC', 'Toyota', 'Corolla', 'Sedán', 'Blanco'),

(2, '2345DEF', 'Peugeot', '308', 'Compacto', 'Negro'),

(3, '3456GHI', 'Renault', 'Clio', 'Utilitario', 'Rojo'),

(4, '4567JKL', 'Seat', 'Ibiza', 'Hatchback', 'Gris'),

(5, '5678MNO', 'Volkswagen', 'Golf', 'Hatchback', 'Azul'),

(6, '6789PQR', 'Ford', 'Focus', 'Sedán', 'Verde'),

(7, '7890STU', 'Opel', 'Corsa', 'Compacto', 'Amarillo'),

(8, '8901VWX', 'Kia', 'Rio', 'Sedán', 'Plateado'),

(9, '9012YZA', 'Hyundai', 'i30', 'Familiar', 'Blanco'),

(10, '0123BCD', 'Citroën', 'C3', 'Compacto', 'Negro');

INSERT INTO MANTENIMIENTO VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10);

INSERT INTO HISTORIAL\_VEHICULO VALUES

(1, 1, 1, 'Cambio de aceite y revisión general', 15000, 'Ninguno'),

(2, 2, 2, 'Cambio de frenos', 20000, 'Ninguno'),

(3, 3, 3, 'Revisión de suspensión', 18000, 'Golpe leve en lateral'),

(4, 4, 4, 'Cambio de batería', 22000, 'Ninguno'),

(5, 5, 5, 'Revisión de neumáticos', 25000, 'Ninguno'),

(6, 6, 6, 'Cambio de filtro de aire', 17000, 'Ninguno'),

(7, 7, 7, 'Cambio de embrague', 30000, 'Pequeño choque trasero'),

(8, 8, 8, 'Cambio de pastillas de freno', 21000, 'Ninguno'),

(9, 9, 9, 'Revisión completa', 35000, 'Abolladura en puerta'),

(10, 10, 10, 'Cambio de aceite', 12000, 'Ninguno');

INSERT INTO SUCURSAL VALUES

(1, 'Sucursal Norte', 'Av. Galicia 1, Lugo'),

(2, 'Sucursal Sur', 'Av. Andalucía 14, Sevilla'),

(3, 'Sucursal Este', 'Calle Levante 9, Valencia'),

(4, 'Sucursal Oeste', 'Calle Poniente 7, Badajoz'),

(5, 'Sucursal Centro', 'Plaza Mayor 3, Madrid'),

(6, 'Sucursal Galicia', 'Rúa da Liberdade 2, Santiago'),

(7, 'Sucursal Cataluña', 'Passeig Gràcia 10, Barcelona'),

(8, 'Sucursal Levante', 'Av. Mediterráneo 22, Alicante'),

(9, 'Sucursal Noroeste', 'Calle Río 6, León'),

(10, 'Sucursal Balear', 'Calle Mar 3, Palma');

INSERT INTO EMPLEADO VALUES

(1, '11111111A', 'Raúl', 'García', 610001001, 1),

(2, '22222222B', 'Sara', 'López', 610001002, 2),

(3, '33333333C', 'Hugo', 'Martínez', 610001003, 3),

(4, '44444444D', 'Julia', 'Navarro', 610001004, 4),

(5, '55555555E', 'Mario', 'Ortega', 610001005, 5),

(6, '66666666F', 'Paula', 'Santos', 610001006, 6),

(7, '77777777G', 'Diego', 'Rey', 610001007, 7),

(8, '88888888H', 'Nerea', 'Muñoz', 610001008, 8),

(9, '99999999I', 'Javier', 'Ibáñez', 610001009, 9),

(10, '00000000J', 'Irene', 'Fernández', 610001010, 10);

INSERT INTO RESERVA VALUES

(1, '2025-06-01', '2025-06-07', 'Confirmada', 1),

(2, '2025-06-10', '2025-06-15', 'Pendiente', 2),

(3, '2025-06-20', '2025-06-25', 'Cancelada', 3),

(4, '2025-07-01', '2025-07-07', 'Confirmada', 4),

(5, '2025-07-10', '2025-07-17', 'Pendiente', 5),

(6, '2025-07-20', '2025-07-27', 'Confirmada', 6),

(7, '2025-08-01', '2025-08-05', 'Confirmada', 7),

(8, '2025-08-10', '2025-08-15', 'Cancelada', 8),

(9, '2025-08-20', '2025-08-25', 'Pendiente', 9),

(10, '2025-09-01', '2025-09-05', 'Confirmada', 10);

INSERT INTO PAGO VALUES

(1, '2025-06-01', 'Tarjeta', 1),

(2, '2025-06-10', 'Transferencia', 2),

(3, '2025-06-20', 'Efectivo', 3),

(4, '2025-07-01', 'Tarjeta', 4),

(5, '2025-07-10', 'Bizum', 5),

(6, '2025-07-20', 'Tarjeta', 6),

(7, '2025-08-01', 'Transferencia', 7),

(8, '2025-08-10', 'Efectivo', 8),

(9, '2025-08-20', 'Tarjeta', 9),

(10, '2025-09-01', 'Bizum', 10);

INSERT INTO FACTURA VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10);

INSERT INTO INCIDENCIA VALUES

(1, 1, 'Retraso en entrega de vehículo'),

(2, 2, 'Problema con el aire acondicionado'),

(3, 3, 'Neumático pinchado'),

(4, 4, 'Cliente insatisfecho con limpieza'),

(5, 5, 'Problema con el sistema de reservas'),

(6, 6, 'Error en facturación'),

(7, 7, 'Vehículo entregado sin repostar'),

(8, 8, 'Problemas con el navegador GPS'),

(9, 9, 'Retraso en devolución de vehículo'),

(10, 10, 'Incidente menor en estacionamiento');

# Funciones y Procedimientos Almacenados

-- Funciones almacenadas

DELIMITER //

CREATE FUNCTION obtener\_edad\_promedio\_clientes() RETURNS DECIMAL(5,2)

DETERMINISTIC

BEGIN

    DECLARE promedio DECIMAL(5,2);

    SELECT AVG(edad) INTO promedio FROM CLIENTE;

    RETURN promedio;

END //

CREATE FUNCTION total\_pagos\_por\_cliente(cliente\_id INT) RETURNS DECIMAL(10,2)

DETERMINISTIC

BEGIN

    DECLARE total DECIMAL(10,2);

    SELECT COUNT(\*) \* 100.00 INTO total

    FROM PAGO P

    JOIN RESERVA R ON P.id\_reserva = R.id\_reserva

    WHERE R.id\_cliente = cliente\_id;

    RETURN total;

END //

-- Procedimientos almacenados

CREATE PROCEDURE insertar\_empleado (

    IN p\_dni VARCHAR(20),

    IN p\_nombre VARCHAR(50),

    IN p\_apellidos VARCHAR(50),

    IN p\_telefono INT,

    IN p\_sucursal INT

)

BEGIN

    INSERT INTO EMPLEADO (id\_empleado, dni, nombre, apellidos, telefono\_empleado, id\_sucursal)

    VALUES ((SELECT IFNULL(MAX(id\_empleado), 0) + 1 FROM EMPLEADO), p\_dni, p\_nombre, p\_apellidos, p\_telefono, p\_sucursal);

END //

CREATE PROCEDURE actualizar\_estado\_reserva (

    IN p\_id\_reserva INT,

    IN p\_nuevo\_estado VARCHAR(20)

)

BEGIN

    UPDATE RESERVA SET estado\_reserva = p\_nuevo\_estado WHERE id\_reserva = p\_id\_reserva;

END //

-- Triggers

-- BEFORE INSERT para comprobar edad mínima

CREATE TRIGGER trg\_before\_insert\_cliente

BEFORE INSERT ON CLIENTE

FOR EACH ROW

BEGIN

    IF NEW.edad < 18 THEN

        SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'El cliente debe ser mayor de edad';

    END IF;

END //

-- AFTER INSERT para log de pagos

CREATE TABLE LOG\_PAGOS (

    id\_log INT AUTO\_INCREMENT PRIMARY KEY,

    id\_pago INT,

    fecha\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TRIGGER trg\_after\_insert\_pago

AFTER INSERT ON PAGO

FOR EACH ROW

BEGIN

    INSERT INTO LOG\_PAGOS (id\_pago) VALUES (NEW.id\_pago);

END //

DELIMITER ;

# Triggers

DELIMITER //

CREATE TRIGGER trg\_before\_insert\_cliente

BEFORE INSERT ON CLIENTE

FOR EACH ROW

BEGIN

    IF NEW.edad < 18 THEN

        SIGNAL SQLSTATE '45000'

        SET MESSAGE\_TEXT = 'El cliente debe ser mayor de edad';

    END IF;

END //

DELIMITER ;

trigers no funcionan..- 21/05/02025

# Consultas SQL

Describe aquí...

# Casos de Prueba y Simulación

Describe aquí...

# Resultados y Verificación

Describe aquí...

# Capturas de Pantalla (opcional)

Describe aquí...

# Conclusiones y Mejoras Futuras

Describe aquí...

# Enlace al Repositorio en GitHub

Describe aquí...