

REPORTE DE PRÁCTICA NO 1.2

Gestión de flotilla de autos

ALUMNO: Esperilla Mendoza Luis Erick
Dr. Eduardo Cornejo-Velázquez



1. Introducción

El diseño e implementación de bases de datos relacionales es un proceso sistemático y metodológico fundamental para la gestión eficiente de la información. A través de una estructura bien definida, se garantiza la integridad, seguridad y optimización del acceso a los datos, permitiendo una administración más efectiva de los recursos.

En esta práctica, se desarrollará un modelo de base de datos en MySQL para la gestión de flotillas de autos, abordando aspectos clave como el control de gastos de gasolina, la regulación de documentos y el mantenimiento de los vehículos. Estos elementos son esenciales para mejorar la operatividad de una flotilla, reducir costos y optimizar el rendimiento de los activos empresariales.

El uso de bases de datos relacionales facilita la organización de la información, permitiendo consultas eficientes y asegurando la trazabilidad de los datos en tiempo real. A través de este ejercicio, se aplicarán principios de modelado de datos, normalización y diseño de esquemas para crear una solución estructurada y funcional.

2. Marco teórico

Para el diseño e implementación de una base de datos relacional orientada a la gestión de flotillas de autos, es fundamental comprender los conceptos clave relacionados con el modelado de bases de datos. En esta sección, se presentan los fundamentos teóricos que sustentan el desarrollo del sistema.

Análisis de requerimientos

El análisis de requerimientos es la fase inicial en el proceso de diseño de bases de datos, donde se identifican las necesidades del sistema y se definen los datos que deben ser almacenados. Este proceso permite establecer las funcionalidades esenciales y las relaciones entre los distintos elementos de la base de datos.

La recopilación de requerimientos puede realizarse mediante entrevistas con usuarios, análisis de sistemas previos y revisión de normativas aplicables.

Modelo Entidad - Relación

El modelo Entidad-Relación (ER) es una representación gráfica utilizada para definir la estructura lógica de una base de datos. Este modelo permite visualizar las entidades (objetos de interés), sus atributos y las relaciones entre ellas.

Utilizar un diagrama ER facilita la normalización de los datos y evita redundancias, asegurando un diseño eficiente y escalable.

Modelo relacional

El modelo relacional es la base teórica sobre la cual operan los sistemas de gestión de bases de datos relacionales (RDBMS). En este modelo, los datos se organizan en tablas conformadas por filas y columnas, donde cada fila representa una instancia de una entidad y cada columna representa un atributo.

La integridad referencial y la normalización son aspectos fundamentales dentro del modelo relacional, ya que garantizan la consistencia y eficiencia en la manipulación de los datos.

SQL

SQL (Structured Query Language) es el lenguaje utilizado para la manipulación y gestión de bases de datos relacionales.

Con SQL, es posible realizar operaciones como la creación de tablas, inserción, modificación y eliminación de datos, así como la realización de consultas complejas mediante la combinación de tablas y funciones agregadas.

3. Herramientas empleadas

- **ERD Plus**

- **Tipo:** Herramienta en línea para diseño de diagramas de entidad-relación (ER).
- **Uso:** Se utiliza para crear diagramas lógicos de bases de datos, definir entidades, relaciones y atributos, y generar scripts SQL automáticamente para crear bases de datos.

- **DataGrip**

- **Tipo:** IDE para bases de datos desarrollado por JetBrains.
- **Uso:** Se utiliza para gestionar y administrar bases de datos, escribir y ejecutar consultas SQL, diseñar esquemas de bases de datos, y conectarse a múltiples servidores de bases de datos, todo en una sola interfaz.

4. Desarrollo

Análisis de requisitos

1. Gestión de Vehículos
 - Registrar, modificar y eliminar vehículos
2. Gestión de conductores
 - Registrar, modificar y eliminar vehiculos
 - Asignar un conductor a un vehículo
3. Gestión de mecánicos y mantenimiento
 - Registrar mecánicos y sus especialidades
 - Registrar mantenimientos realizados a los vehículos
4. Gestión de combustible
 - Registrar carga de de combustible de los vehiculos
 - Calcular el costo total de cada carga de combustible

Modelo Entidad - Relación

En la Figura 1 se presenta la propuesta de Modelo Entidad - Relación para el caso de la Flotilla de autos

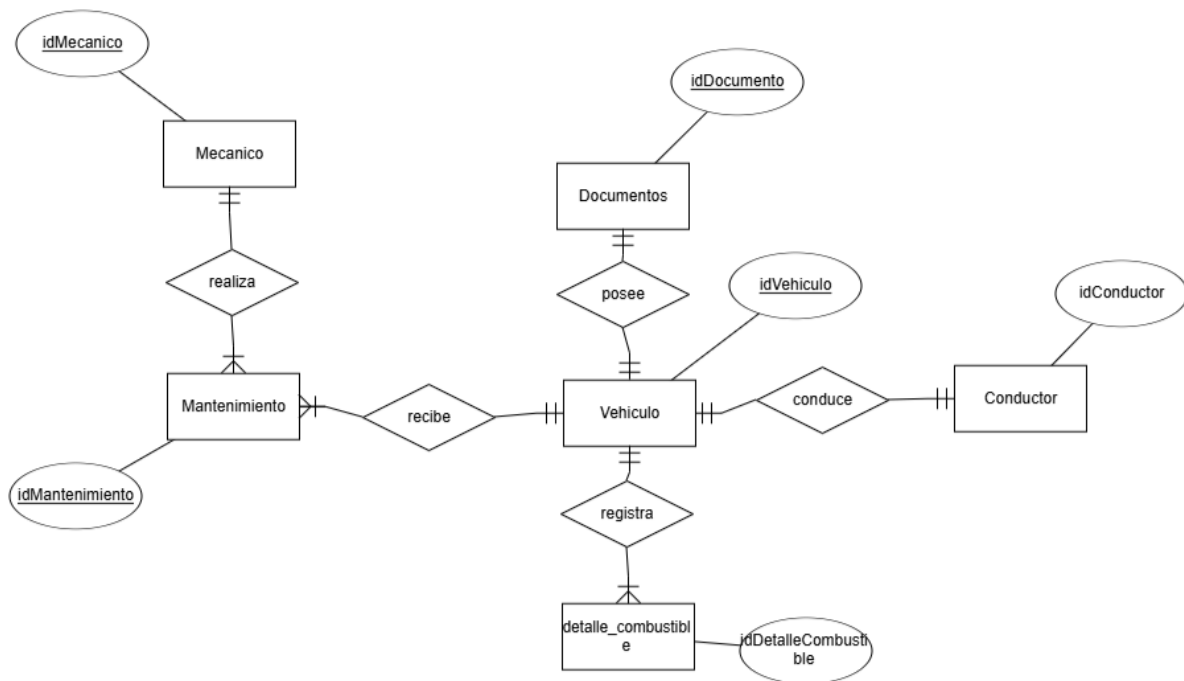


Figure 1: Modelo Entidad - Relación propuesto.

Modelo relacional

En la Figura 2 se presenta la propuesta de Modelo Entidad - Relación para el caso de la Flotilla de autos

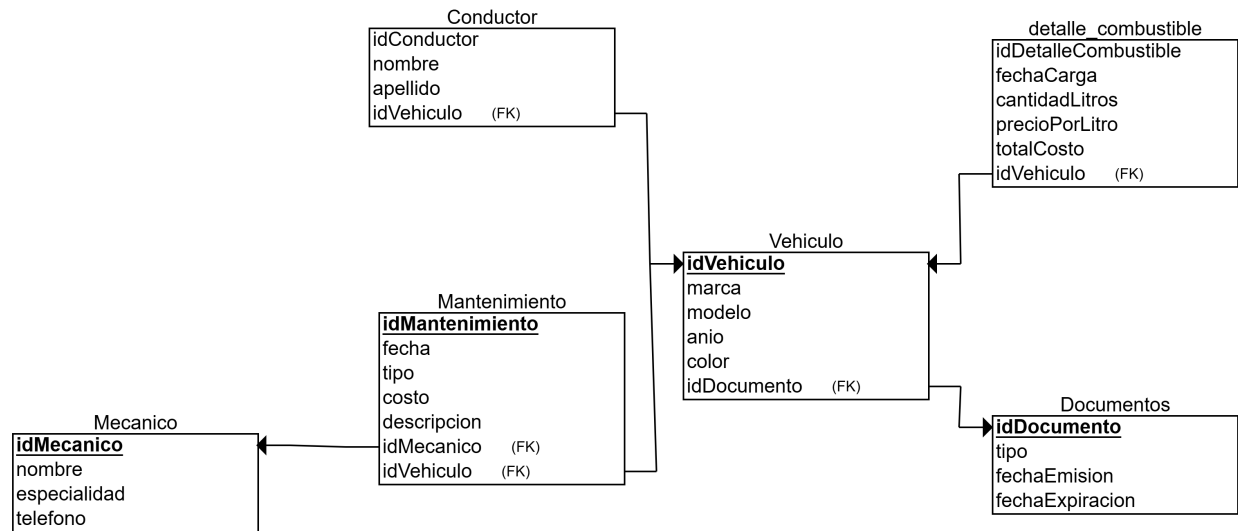


Figure 2: Modelo Relacional propuesto.

Sentencias SQL

En el Listado 1 se presenta la sentencia SQL para crear la base de datos flotillaAutos.

Listing 1: Crear base de datos flotillaAutos.

```
CREATE DATABASE flotillaAutos ;
```

En el listado 2 se presenta la sentencia SQL para utilizar la base de datos

Listing 2: Utilizar base de datos flotillaAutos.

```
USE flotillaAutos ;
```

En el listado 3 se presenta la sentencia SQL para crear la tabla vehiculo

Listing 3: Crear la tabla vehiculo en la base de datos.

```
CREATE TABLE Vehiculo (
    idVehiculo INT NOT NULL,
    marca VARCHAR(80) NOT NULL,
    modelo VARCHAR(80) NOT NULL,
    anio INT NOT NULL,
    color VARCHAR(80) NOT NULL,
    idDocumento INT NOT NULL,
    idConductor INT,
    PRIMARY KEY (idVehiculo), FOREIGN KEY (idDocumento) REFERENCES Documentos(idDocumento),
    FOREIGN KEY (idConductor) REFERENCES Conductor(idConductor)
);
```

En el listado 4 se presenta la sentencia SQL para crear la tabla documentos

Listing 4: Crear la tabla documentos en la base de datos.

```
CREATE TABLE Documentos (
    idDocumento INT NOT NULL,
    tipo VARCHAR(80) NOT NULL,
    fechaEmision DATE NOT NULL,
    fechaExpiracion DATE NOT NULL,
```

```
PRIMARY KEY (idDocumento)
);
```

En el listado 5 se presenta la sentencia SQL para crear la tabla conductor

Listing 5: Crear la tabla conductor en la base de datos.

```
CREATE TABLE Conductor (
  idConductor INT NOT NULL,
  nombre VARCHAR(80) NOT NULL,
  apellido VARCHAR(80) NOT NULL,
PRIMARY KEY (idConductor)
);
```

En el listado 6 se presenta la sentencia SQL para crear la tabla mecanico

Listing 6: Crear la tabla mecanico en la base de datos.

```
CREATE TABLE Mecanico (
  idMecanico INT NOT NULL,
  nombre VARCHAR(80) NOT NULL,
  especialidad VARCHAR(80) NOT NULL,
  telefono VARCHAR(15) NOT NULL,
PRIMARY KEY (idMecanico)
);
```

En el listado 7 se presenta la sentencia SQL para crear la tabla detalleCombustible

Listing 7: Crear la tabla detalleCombustible en la base de datos.

```
CREATE TABLE detalle_combustible (
  idDetalleCombustible INT NOT NULL,
  fechaCarga DATE NOT NULL,
  cantidadLitros FLOAT NOT NULL,
  precioPorLitro FLOAT NOT NULL,
  totalCosto FLOAT NOT NULL,
  idVehiculo INT NOT NULL,
PRIMARY KEY (idDetalleCombustible),
FOREIGN KEY (idVehiculo) REFERENCES Vehiculo(idVehiculo)
);
```

En el listado 8 se presenta la sentencia SQL para crear la tabla mantenimiento

Listing 8: Crear la tabla mantenimiento en la base de datos.

```
CREATE TABLE Mantenimiento (
  idMantenimiento INT NOT NULL,
  fecha DATE NOT NULL,
  tipo VARCHAR(80) NOT NULL,
  costo FLOAT NOT NULL,
  descripcion VARCHAR(255) NOT NULL,
  idMecanico INT NOT NULL,
  idVehiculo INT NOT NULL,
PRIMARY KEY (idMantenimiento),
FOREIGN KEY (idMecanico) REFERENCES Mecanico(idMecanico),
FOREIGN KEY (idVehiculo) REFERENCES Vehiculo(idVehiculo)
);
```

En el listado 9 se presenta la sentencia SQL para insertar datos en la tabla mecanico

Listing 9: Insertar datos en la tabla mecanico en la base de datos.

```
INSERT INTO Mecanico (idMecanico, nombre, especialidad, telefono) VALUES
(1, 'Luis-Martinez', 'Motores', '555-1234'),
(2, 'Sofia-Gomez', 'Frenos', '555-5678'),
(3, 'Ricardo-Torres', 'Suspension', '555-9101'),
(4, 'Elena-Ruiz', 'Electrico', '555-1122'),
(5, 'Jorge-Herrera', 'Transmision', '555-3344');
```

En el listado 10 se presenta la sentencia SQL para insertar datos en la tabla conductor

Listing 10: Insertar datos en la tabla conductor en la base de datos.

```
INSERT INTO Conductor (idConductor, nombre, apellido) VALUES
(1, 'Juan', 'Perez'),
(2, 'Maria', 'Lopez'),
(3, 'Carlos', 'Garcia'),
(4, 'Ana', 'Rodriguez'),
(5, 'Pedro', 'Fernandez');
```

En el listado 11 se presenta la sentencia SQL para insertar datos en la tabla documentos

Listing 11: Insertar datos en la tabla documentos en la base de datos.

```
INSERT INTO Documentos (idDocumento, tipo, fechaEmision, fechaExpiracion)
VALUES
(1, 'Licencia-de-Conducir', '2023-01-15', '2028-01-15'),
(2, 'Tarjeta-de-Circulacion', '2022-06-10', '2027-06-10'),
(3, 'Seguro-Vehicular', '2024-02-01', '2025-02-01'),
(4, 'Revision-Tecnica', '2023-05-20', '2024-05-20'),
(5, 'Permiso-Especial', '2023-08-30', '2024-08-30');
```

En el listado 12 se presenta la sentencia SQL para insertar datos en la tabla vehiculo

Listing 12: Insertar datos en la tabla vehiculo en la base de datos.

```
INSERT INTO Vehiculo (idVehiculo, marca, modelo, anio, color, idDocumento,
idConductor) VALUES
(1, 'Toyota', 'Corolla', 2020, 'Rojo', 1, 1),
(2, 'Honda', 'Civic', 2019, 'Azul', 2, 2),
(3, 'Ford', 'Focus', 2021, 'Negro', 3, 3),
(4, 'Chevrolet', 'Malibu', 2018, 'Blanco', 4, 4),
(5, 'Nissan', 'Sentra', 2022, 'Gris', 5, 5);
```

En el listado 13 se presenta la sentencia SQL para insertar datos en la tabla detalleCombustible

Listing 13: Insertar datos en la tabla detalleCombustible en la base de datos.

```
INSERT INTO detalle_combustible (idDetalleCombustible, fechaCarga, cantidadLitros,
precioPorLitro,
totalCosto, idVehiculo)
VALUES
(1, '2024-02-01', 40, 1.50, 60, 1),
(2, '2024-02-05', 35, 1.55, 54.25, 2),
(3, '2024-02-10', 50, 1.45, 72.50, 3),
(4, '2024-02-15', 38, 1.60, 60.80, 4),
(5, '2024-02-20', 45, 1.52, 68.40, 5);
```

En el listado 13 se presenta la sentencia SQL para insertar datos en la tabla mantenimiento

Listing 14: Insertar datos en la tabla mantenimiento en la base de datos.

```
INSERT INTO Mantenimiento (idMantenimiento, fecha, tipo, costo, descripcion, idMec.
```



```
(1, '2024-01-15', 'Cambio-de-aceite', 50.00, 'Cambio-de-aceite-y-filtro', 1, 1),
(2, '2024-01-20', 'Revision-de-frenos',
80.00, 'Cambio-de-pastillas-de-freno', 2, 2),
(3, '2024-01-25', 'Alineacion-y-balanceo',
60.00, 'Alineacion-de-ruedas-y-balanceo', 3, 3),
(4, '2024-02-01', 'Revision-electrica',
90.00, 'Diagnostico-del-sistema-electrico', 4, 4),
(5, '2024-02-05', 'Cambio-de-transmision',
500.00, 'Reemplazo-de-transmision-automatica', 5, 5);
```

5. Conclusiones

Durante el desarrollo de esta practica pude fortalecer mis habilidades en el diseño de bases de datos, específicamente con los esquemas de entidad-relación y posteriormente al esquema relacional.

Este proceso me ayudo a comprender de mejor manera la importancia de una base de datos bien estructurada.

Así mismo reforcé el como crear las sentencias en SQL para poder crear tablas y también para poder insertar datos.

En general, este proyecto no solo reforzó mis conocimientos teóricos sobre bases de datos, sino que también me brindó experiencia práctica en su implementación. Me siento más preparado para abordar futuros desafíos relacionados con el diseño y gestión de bases de datos en sistemas reales.

Referencias Bibliográficas

References

- [1] Edenred, E. (2023, 17 febrero). Flotilla de autos: cómo administrarla. Edenred.
<https://www.edenred.mx/blog/flotilla-de-autos-como-administrarla>