

REPORTE DE PRÁCTICA NO. 1.2

NOMBRE DE LA PRÁCTICA: Gestion de Flotilla de Autos

ALUMNO: Saúl Jiménez Mercado
Dr. Eduardo Cornejo-Velázquez



1. Introducción

En esta práctica se busca dar el manejo adecuado de una flotilla de autos, la cual necesita de una administración muy precisa de todo lo que ocurre en cada uno de los autos dentro de la flotilla. El uso de una base de datos para este tipo de servicios funciona como una herramienta esencial para la gestión eficiente y organizada de los vehículos, con el objetivo de optimizar su uso y mantenimiento.

La base de datos facilita la toma de decisiones informadas respecto a la planificación de mantenimientos, el uso de la gasolina y que el personal esté bien capacitado. Con una buena administración, se pueden reducir costos, mejorar la seguridad vehicular y aumentar la eficiencia operativa de la empresa.

2. Marco teórico

Análisis de requerimientos

El análisis de requerimientos es una fase crucial en el desarrollo de software y otros proyectos, donde se identifican, documentan y validan las necesidades y expectativas de los clientes o usuarios finales. Este proceso es fundamental para asegurar que el producto final cumpla con las expectativas del cliente y satisfaga sus necesidades.

En el caso de la administración de flotillas, es importante identificar los elementos clave a gestionar, como los vehículos, conductores, mantenimientos, consumos de combustible y capacitaciones del personal.

Modelo Entidad - Relación

El modelo entidad-relación es una herramienta que permite representar de manera simplificada cómo personas, objetos o conceptos se relacionan entre sí. Se utiliza para exponer cómo se organiza la información en una base de datos. Se emplea para diseñar bases de datos en múltiples campos, como los sistemas de información empresarial, los procesos de negocio, la investigación, la educación o la ingeniería de software.

En la administración de flotillas, este modelo permite visualizar las relaciones entre los elementos esenciales, como la empresa, los vehículos, los conductores y las rutas asignadas.

Modelo relacional

El modelo de datos relacional proporcionó una forma estándar de representar y consultar datos que podría utilizar cualquier aplicación. Desde el principio, los desarrolladores se dieron cuenta de que la virtud principal del modelo de base de datos relacional era el uso de tablas, ya que era una forma intuitiva, eficiente y flexible de almacenar y acceder a información estructurada.

En el caso de flotillas, permite almacenar información estructurada en diferentes tablas, asegurando integridad y consistencia en los datos.

SQL

El lenguaje de consulta estructurada (SQL) es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos. Se pueden usar instrucciones SQL para almacenar, actualizar, eliminar, buscar y recuperar información de la base de datos. También permite mantener y optimizar el rendimiento de la base de datos. En la gestión de flotillas, SQL permite realizar consultas eficientes para obtener información relevante sobre los vehículos, conductores y mantenimiento de los mismos..

3. Herramientas empleadas

Describir qué herramientas se han utilizado...

1. ERD Plus: ERD Plus es una herramienta de diagramas ER en línea que permite dibujar todos los niveles de modelos de datos. Puede utilizar notaciones como pata de gallo y le permite generar archivos DDL de SQL para crear su base de datos física a partir de un modelo de datos físico. ERDPlus es compatible con muchos sistemas de gestión de bases de datos, como MySQL, SQL Server, Oracle e IBM DB2. Al igual que otras herramientas ERD en línea, pone a disposición sus modelos de datos desde cualquier lugar, en cualquier dispositivo o plataforma.
2. MySQL Server: MySQL Workbench es una herramienta visual unificada para arquitectos de bases de datos, desarrolladores y administradores de bases de datos. MySQL Workbench proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para configuración de servidores, administración de usuarios, copias de seguridad y mucho más. Permite diseñar, visualizar y mantener esquemas de bases de datos de manera eficiente.

Con el uso de estas herramientas, se puede estructurar una base de datos optimizada para la administración de flotillas, asegurando un mejor control sobre los vehículos y recursos asociados.

4. Desarrollo

Análisis de requisitos

Modelo Entidad - Relación

En la Tabla 1 se presenta la propuesta de Modelo Entidad - Relación para una flotilla de autos

Table 1: Matriz de realaciones.

Entidades	Auto	Gasolina	Documentos	Capacitacion	Conductor	Mantenimiento	Piezas	Ruta
Auto	X	Ocupa	Tiene	X	Utiliza	Recibe	X	Realiza
Gasolina	Ocupa	X	X	X	X	X	X	X
Documentos	Tiene	X	X	X	X	X	X	X
Capacitacion	X	X	X	X	Recibe	X	X	X
Conductor	X	X	X	Recibe	X	X	X	X
Mantenimineto	Recibe	X	X	X	X	X	Utiliza	X
Piezas	X	X	X	X	X	Utiliza	X	X
Ruta	Realiza	X	X	X	X	X	X	X

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

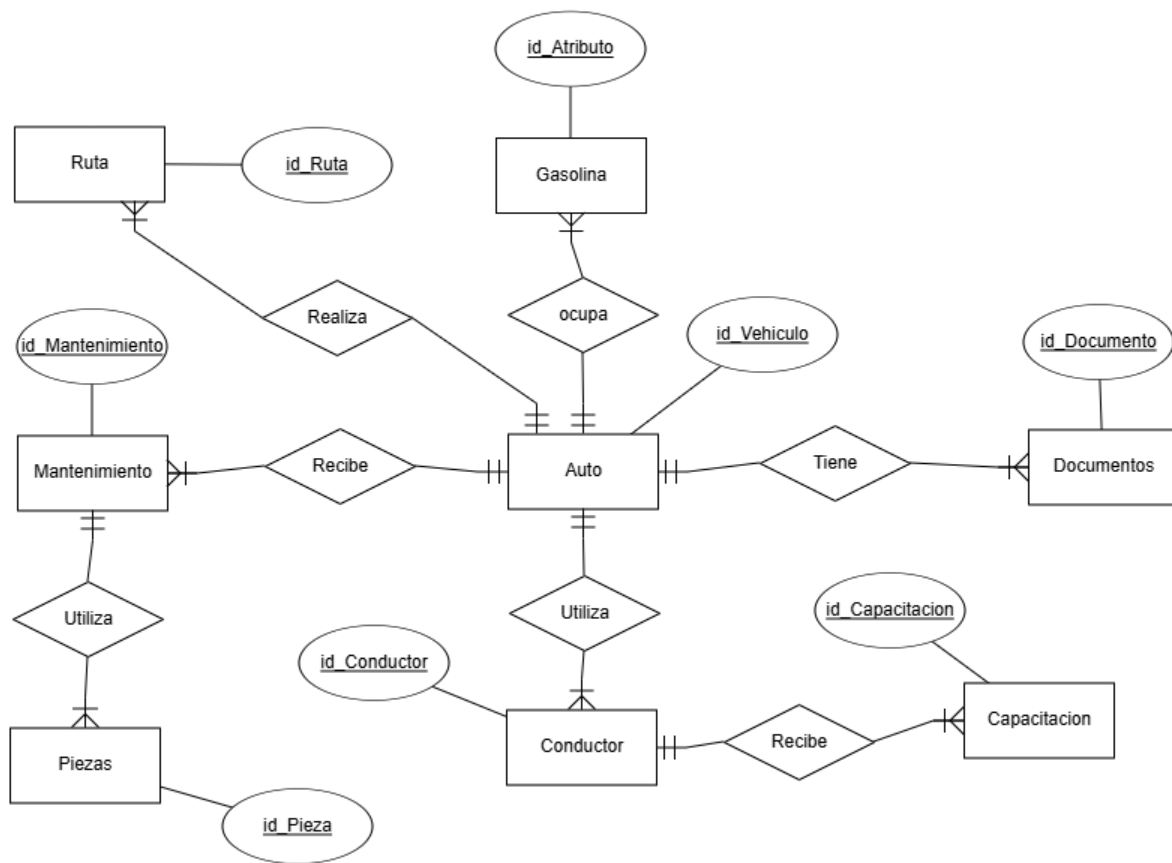


Figure 1: Modelo Entidad - Relación propuesto.

Modelo relacional

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

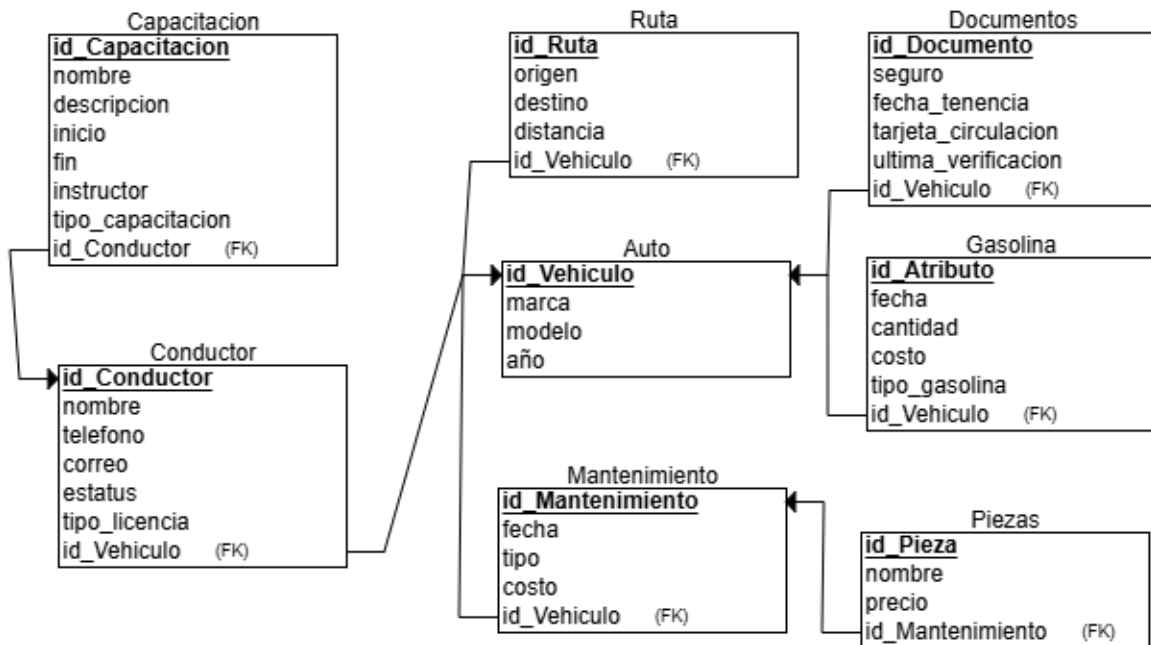


Figure 2: Modelo Relacional propuesto.

Sentencias SQL

Presentar las sentencias para crear la base de datos y tablas. Además incluir las sentencias para insertar registros.

En el Listado 1 se presenta la sentencia SQL para crear la base de datos de una flota de autos.

Listing 1: Crear base de datos flota de autos.

```

CREATE DATABASE flotillaautos;
use flotillaautos;

CREATE TABLE Auto
(
  id_Vehiculo INT NOT NULL,
  marca VARCHAR(80) NOT NULL,
  modelo VARCHAR(80) NOT NULL,
  ano DATE NOT NULL,
  PRIMARY KEY (id_Vehiculo)
);

CREATE TABLE Gasolina
(
  id_Atributo INT NOT NULL,
  fecha DATE NOT NULL,
  cantidad INT NOT NULL,
  costo INT NOT NULL,
  tipo_gasolina VARCHAR(80) NOT NULL,
  id_Vehiculo INT NOT NULL,
  PRIMARY KEY (id_Atributo),

```

```
FOREIGN KEY (id_Vehiculo) REFERENCES Auto(id_Vehiculo)
);
```

```
CREATE TABLE Documentos
```

```
(
    id_Documento INT NOT NULL,
    seguro VARCHAR(80) NOT NULL,
    fecha_tenencia DATE NOT NULL,
    tarjeta_circulacion VARCHAR(80) NOT NULL,
    ultima_verificacion DATE NOT NULL,
    id_Vehiculo INT NOT NULL,
    PRIMARY KEY (id_Documento),
    FOREIGN KEY (id_Vehiculo) REFERENCES Auto(id_Vehiculo)
);
```

```
CREATE TABLE Conductor
```

```
(
    id_Conductor INT NOT NULL,
    nombre VARCHAR(80) NOT NULL,
    telefono VARCHAR(80) NOT NULL,
    correo VARCHAR(80) NOT NULL,
    estatus VARCHAR(80) NOT NULL,
    tipo_licencia VARCHAR(80) NOT NULL,
    id_Vehiculo INT NOT NULL,
    PRIMARY KEY (id_Conductor),
    FOREIGN KEY (id_Vehiculo) REFERENCES Auto(id_Vehiculo)
);
```

```
CREATE TABLE Mantenimiento
```

```
(
    id_Mantenimiento INT NOT NULL,
    fecha DATE NOT NULL,
    tipo VARCHAR(80) NOT NULL,
    costo INT NOT NULL,
    id_Vehiculo INT NOT NULL,
    PRIMARY KEY (id_Mantenimiento),
    FOREIGN KEY (id_Vehiculo) REFERENCES Auto(id_Vehiculo)
);
```

```
CREATE TABLE Capacitacion
```

```
(
    id_Capacitacion INT NOT NULL,
    nombre VARCHAR(80) NOT NULL,
    descripcion VARCHAR(80) NOT NULL,
    inicio DATE NOT NULL,
    fin DATE NOT NULL,
    instructor VARCHAR(80) NOT NULL,
    tipo_capacitacion VARCHAR(80) NOT NULL,
    id_Conductor INT NOT NULL,
    PRIMARY KEY (id_Capacitacion),
    FOREIGN KEY (id_Conductor) REFERENCES Conductor(id_Conductor)
);
```

```
CREATE TABLE Piezas
```

```
(
    id_Pieza INT NOT NULL,
    nombre VARCHAR(80) NOT NULL,
    precio INT NOT NULL,
    id_Mantenimiento INT NOT NULL,
    PRIMARY KEY (id_Pieza),
    FOREIGN KEY (id_Mantenimiento) REFERENCES Mantenimiento(id_Mantenimiento)
);
```

CREATE TABLE Ruta

```
(
    id_Ruta INT NOT NULL,
    origen VARCHAR(80) NOT NULL,
    destino VARCHAR(80) NOT NULL,
    distancia INT NOT NULL,
    id_Vehiculo INT NOT NULL,
    PRIMARY KEY (id_Ruta),
    FOREIGN KEY (id_Vehiculo) REFERENCES Auto(id_Vehiculo)
);
```

use flotillaautos;

INSERT INTO Auto (id_Vehiculo, marca, modelo, año) **VALUES**

```
(1, 'Toyota', 'Corolla', '2020-01-01'),
(2, 'Honda', 'Civic', '2019-01-01'),
(3, 'Ford', 'Focus', '2018-01-01'),
(4, 'Nissan', 'Sentra', '2021-01-01'),
(5, 'Chevrolet', 'Malibu', '2022-01-01');
```

INSERT INTO Gasolina (id_Atributo, fecha, cantidad, costo, tipo_gasolina, id_Vehiculo)

```
(1, '2024-02-01', 40, 800, 'Premium', 1),
(2, '2024-02-02', 35, 700, 'Regular', 2),
(3, '2024-02-03', 50, 1000, 'Diesel', 3),
(4, '2024-02-04', 45, 900, 'Premium', 4),
(5, '2024-02-05', 38, 760, 'Regular', 5);
```

INSERT INTO Documentos (id_Documento, seguro, fecha_tenencia, tarjeta_circulacion)

```
(1, 'GNP', '2023-01-01', 'ABC123', '2023-06-01', 1),
(2, 'AXA', '2023-02-01', 'DEF456', '2023-07-01', 2),
(3, 'Quilintas', '2023-03-01', 'GHI789', '2023-08-01', 3),
(4, 'Mapfre', '2023-04-01', 'JKL012', '2023-09-01', 4),
(5, 'HDI', '2023-05-01', 'MNO345', '2023-10-01', 5);
```

INSERT INTO Conductor (id_Conductor, nombre, telefono, correo, estatus, tipo_licencia)

```
(1, 'Carlos Pérez', '5523456789', 'carlos@example.com', 'Activo', 'A', 1),
(2, 'Ana López', '5534567890', 'ana@example.com', 'Activo', 'B', 2),
(3, 'Miguel Torres', '5545678901', 'miguel@example.com', 'Inactivo', 'C', 3),
(4, 'Laura Ramírez', '5556789012', 'laura@example.com', 'Activo', 'A', 4),
(5, 'Javier Gómez', '5567890123', 'javier@example.com', 'Suspendido', 'B', 5);
```

INSERT INTO Mantenimiento (id_Mantenimiento, fecha, tipo, costo, id_Vehiculo)

```
(1, '2024-01-15', 'Cambio-de-aceite', 500, 1),
(2, '2024-01-20', 'Alineación', 700, 2),
(3, '2024-01-25', 'Frenos', 1200, 3),
```

```
(4, '2024-02-01', 'Cambio-de-llantas', 2500, 4),
(5, '2024-02-10', 'Batería', 1800, 5);
```

```
INSERT INTO Capacitacion (id_Capacitacion, nombre, descripcion, inicio, fin, instructor)
VALUES
(1, 'Manejo-Defensivo', 'Técnicas-para-evitar-accidentes', '2024-01-10', '2024-01-15', 'Juan-Pérez'),
(2, 'Primeros-Auxilios', 'Atención-en-emergencias-viales', '2024-02-05', '2024-02-10', 'María-García'),
(3, 'Conducción-Eficiente', 'Optimización-del-consumo-de-combustible', '2024-03-01', '2024-03-05', 'Carlos-Rodríguez'),
(4, 'Normativa-de-Tránsito', 'Reglas-actualizadas-de-tráfico', '2024-04-12', '2024-04-18', 'Ana-López'),
(5, 'Manejo-en-Climas-Extremos', 'Técnicas-para-conducir-en-lluvia-y-nieve', '2024-05-01', '2024-05-05', 'Diego-Martínez');
```

```
INSERT INTO Piezas (id_Pieza, nombre, precio, id_Mantenimiento) VALUES
(1, 'Filtro-de-Aceite', 150, 1),
(2, 'Balanceo-de-Llantas', 600, 2),
(3, 'Pastillas-de-Freno', 900, 3),
(4, 'Neumático', 2000, 4),
(5, 'Batería-de-Auto', 1800, 5);
```

```
INSERT INTO Ruta (id_Ruta, origen, destino, distancia, id_Vehiculo) VALUES
(1, 'Ciudad-de-México', 'Guadalajara', 550, 1),
(2, 'Monterrey', 'Saltillo', 90, 2),
(3, 'Puebla', 'Veracruz', 280, 3),
(4, 'Querétaro', 'San-Luis-Potosí', 180, 4),
(5, 'Mérida', 'Cancún', 300, 5);
```

5. Conclusiones

La administración de flotas de autos es un proceso complejo que requiere herramientas eficientes para la gestión de datos. El uso de bases de datos relacionales y modelos entidad-relación facilita la organización y el acceso a la información, permitiendo una toma de decisiones más informada. Gracias a herramientas como ERD Plus y MySQL Workbench, es posible diseñar y mantener estructuras de datos optimizadas, mejorando la eficiencia operativa de la empresa. Con una administración adecuada, se pueden reducir costos, optimizar el mantenimiento de los vehículos y garantizar la seguridad del personal y los activos de la organización.

Referencias Bibliográficas

References

- [1] ¿Que es SQL? - Explicación de lenguaje de consulta estructurado (SQL) - AWS.(s.f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/sql/>
- [2] Edenred, E. (2022, diciembre 2). Flotilla de autos: cómo administrarla. Edenred.mx. <https://www.edenred.mx/blog/flotilla-de-autos-como-administrarla>