

# REPORTE DE PRÁCTICA NO 1.2

**NOMBRE DE LA PRÁCTICA:** Gestion de Flotilla de Autos

**ALUMNOS:** Jordan Gael Sosa De la Cruz, Hassiel Camacho Meneses  
Dr. Eduardo Cornejo-Velázquez



## 1. Introducción

La práctica muestra la importancia de aplicar un proceso sistemático y metodológico en el diseño de bases de datos, ya que esto permite organizar y administrar de mejor manera la información relacionada con una flotilla de autos. Siguiendo un orden lógico paso a paso, mientras que también implica usar técnicas y métodos probados para asegurar que el diseño sea correcto y eficiente. Con un modelo bien estructurado es posible llevar un control claro de los vehículos, los conductores, el mantenimiento y los gastos, lo que facilita la toma de decisiones y mejora la gestión.



Figure 1: Flotilla de vehículos

## 2. Marco teórico

La gestión de flotillas de autos consiste en organizar y supervisar todos los aspectos relacionados con los vehículos de una empresa. Una administración adecuada ayuda a reducir gastos, mejorar la eficiencia y prolongar la vida útil de los vehículos.

El diseño de una bases de datos relacionales es importante para la gestión de flotillas, ya que permite almacenar y relacionar información de manera ordenada, evitando duplicidad de datos y facilitando consultas rápidas y precisas.

### Análisis de requerimientos

#### - Requerimientos Funcionales

##### 1. Gestión de Automóviles

- Registrar los automóviles con sus atributos: `id_automovil`, `modelo`, `año`, `placa`, `marca`, `estado`, `kilometraje`, `tipo`.
- Cada automóvil cuenta con documentos asociados (tenencia, seguro, verificación, tarjeta de circulación).
- Posibilidad de consultar el estado de cada automóvil (vigencias de documentos, kilometraje, etc.).

##### 2. Gestión de Conductores

- Registrar conductores con sus atributos: `id_conductor`, `nombre`, `telefono`, `correo`.
- Cada conductor está asignado a un automóvil.
- Cada conductor posee documentos: licencia vigente, seguro médico, etc.

##### 3. Gestión de Mantenimientos

- Registrar mantenimientos realizados con atributos: `id_mantenimiento`, `fecha`, `descripcion`, `costo`.
- Asociar cada mantenimiento a un automóvil.
- Posibilidad de generar un reporte del mantenimiento.

##### 4. Gestión de Daños

- Registrar daños de los automóviles con atributos: `id_daño`, `descripcion`, `fecha`, `monto`.
- Los daños se reportan y quedan ligados a un automóvil y un conductor.

##### 5. Gestión de Reportes

- Generar reportes relacionados con:
  - Mantenimientos
  - Daños
  - Gastos
- Los reportes están ligados a automóviles y conductores.

##### 6. Gestión de Gastos

- Registrar gastos como gasolina, monto, fecha.
- Asociar los gastos con un reporte, automóvil y conductor.
- Permitir calcular el costo total de operación de un automóvil/conductor.

## - Requerimientos No Funcionales

### 1. Integridad de datos

- Uso de llaves primarias (`id_automovil`, `id_conductor`, etc.).
- Relaciones bien definidas entre entidades para evitar datos huérfanos.

### 2. Seguridad

- Protección de la información sensible de conductores (documentos, seguros, datos de contacto).
- Control de accesos según roles (administrador, conductor, supervisor).

### 3. Mantenibilidad Base de datos modular: entidades separadas para documentos, gastos, reportes, etc.

### 4. Escalabilidad El modelo soporta la incorporación de nuevos automóviles, conductores y registros de mantenimiento sin modificar la estructura principal.

[?]

## Modelo Entidad - Relación

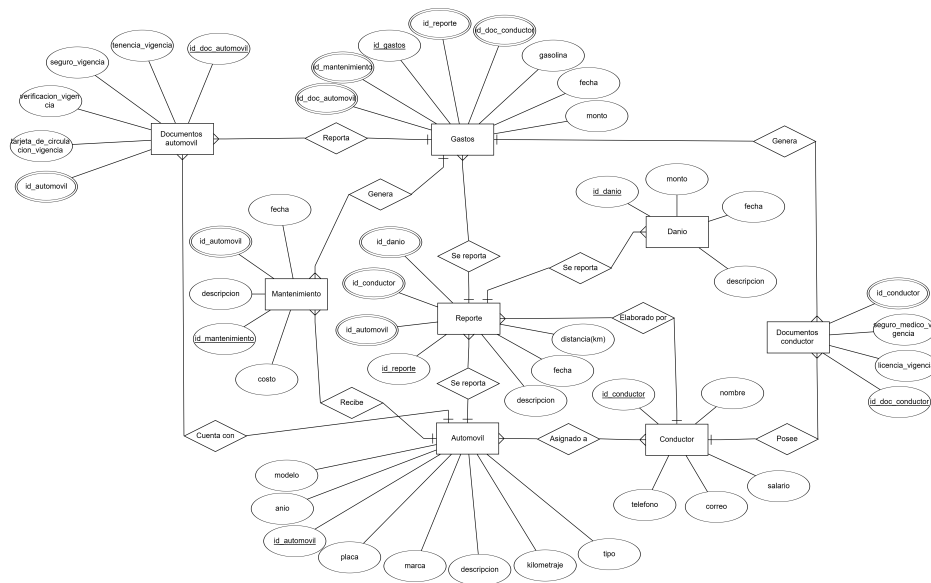


Figure 2: Entidad - Relación

Modelo relacional

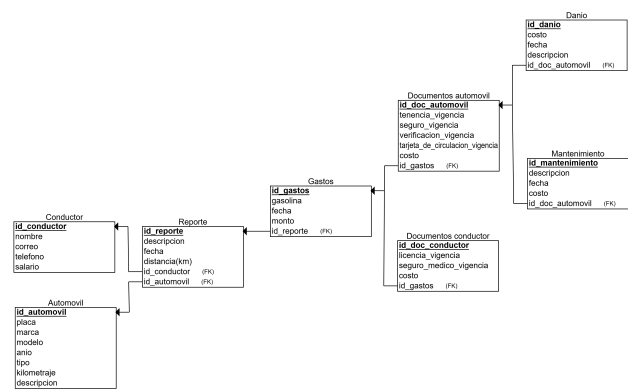


Figure 3: Relacional

## SQL

Listing 1: Descripción de comandos SQL

```
CREATE DATABASE flotilla ;

USE flotilla ;

CREATE TABLE Automovil (
    id_automovil INT NOT NULL,
    placa VARCHAR(10) NOT NULL,
    marca VARCHAR(250) NOT NULL,
    modelo VARCHAR(250) NOT NULL,
    anio INT NOT NULL,
    tipo VARCHAR(250) NOT NULL,
    kilometraje FLOAT NOT NULL,
    descripcion VARCHAR(250) NOT NULL,
    PRIMARY KEY (id_automovil)
);

CREATE TABLE Conductor (
    id_conductor INT NOT NULL,
    nombre VARCHAR(250) NOT NULL,
    correo VARCHAR(250) NOT NULL,
    telefono VARCHAR(15) NOT NULL,
    salario FLOAT NOT NULL,
    PRIMARY KEY (id_conductor)
);

CREATE TABLE Reporte (
    id_reporte INT NOT NULL,
    descripcion VARCHAR(250) NOT NULL,
    fecha DATE NOT NULL,
    distancia_km INT NOT NULL,
    id_conductor INT NOT NULL,
    id_automovil INT NOT NULL,
    PRIMARY KEY (id_reporte),
    FOREIGN KEY (id_conductor) REFERENCES Conductor(id_conductor),
    FOREIGN KEY (id_automovil) REFERENCES Automovil(id_automovil)
);

CREATE TABLE Gastos (
    id_gastos INT NOT NULL,
    gasolina INT NOT NULL,
    fecha DATE NOT NULL,
    monto INT NOT NULL,
    id_reporte INT NOT NULL,
    PRIMARY KEY (id_gastos),
    FOREIGN KEY (id_reporte) REFERENCES Reporte(id_reporte)
);

CREATE TABLE Documentos_automovil (
    id_doc_automovil INT NOT NULL,
    tenencia_vigencia DATE NOT NULL,
    seguro_vigencia DATE NOT NULL,
```

```
    verificacion_vigencia DATE NOT NULL,  
    tarjeta_circulacion_vigencia DATE NOT NULL,  
    costo INT NOT NULL,  
    id_gastos INT NOT NULL,  
    PRIMARY KEY (id_doc_automovil),  
    FOREIGN KEY (id_gastos) REFERENCES Gastos(id_gastos)  
);
```

```
CREATE TABLE Mantenimiento (  
    id_mantenimiento INT NOT NULL,  
    descripcion VARCHAR(250) NOT NULL,  
    fecha DATE NOT NULL,  
    costo INT NOT NULL,  
    id_doc_automovil INT NOT NULL,  
    PRIMARY KEY (id_mantenimiento),  
    FOREIGN KEY (id_doc_automovil) REFERENCES Documentos_automovil(id_doc_automovil)  
);
```

```
CREATE TABLE Documentos_conductor (  
    id_doc_conductor INT NOT NULL,  
    licencia_vigencia DATE NOT NULL,  
    seguro_medico_vigencia DATE NOT NULL,  
    costo INT NOT NULL,  
    id_gastos INT NOT NULL,  
    PRIMARY KEY (id_doc_conductor),  
    FOREIGN KEY (id_gastos) REFERENCES Gastos(id_gastos)  
);
```

```
CREATE TABLE Danio (  
    id_danio INT NOT NULL,  
    costo INT NOT NULL,  
    fecha DATE NOT NULL,  
    descripcion VARCHAR(250) NOT NULL,  
    id_doc_automovil INT NOT NULL,  
    PRIMARY KEY (id_danio),  
    FOREIGN KEY (id_doc_automovil) REFERENCES Documentos_automovil(id_doc_automovil)  
);
```

### 3. Herramientas empleadas

**ERDPlus:** usamos esta herramienta en línea para diseñar el diagrama entidad-relación y el modelo relacional. Nos sirvió para organizar y representar de manera gráfica las entidades, atributos y relaciones de la base de datos antes de implementarla. Fue muy útil porque nos dio una visión clara y ordenada del sistema, así como también nos ayudó a generar las líneas de código a usar en SQL.

**MySQL Server:** lo usamos para implementar el modelo diseñado en ERDPlus. Aquí creamos las tablas, las llaves primarias y foráneas, y ejecutamos consultas para comprobar que las relaciones funcionaran correctamente. Esta herramienta nos permitió validar que todo el modelo estuviera bien estructurado.

### 4. Desarrollo SQL

```
mysql> show tables;
+-----+
| Tables_in_flotilla |
+-----+
| automovil          |
| conductor          |
| danio              |
| documentos_automovil |
| documentos_conductor |
| gastos             |
| mantenimiento     |
| reporte            |
+-----+
8 rows in set (0.00 sec)
```

Figure 4: SHOW TABLES;



```
mysql> describe automovil;
```

Field	Type	Null	Key	Default	Extra
id_automovil	int	NO	PRI	NULL	
placa	varchar(10)	NO		NULL	
marca	varchar(250)	NO		NULL	
modelo	varchar(250)	NO		NULL	
anio	int	NO		NULL	
tipo	varchar(250)	NO		NULL	
kilometraje	float	NO		NULL	
descripcion	varchar(250)	NO		NULL	

```
8 rows in set (0.03 sec)
```

```
mysql> DESCRIBE Conductor;
```

Field	Type	Null	Key	Default	Extra
id_conductor	int	NO	PRI	NULL	
nombre	varchar(250)	NO		NULL	
correo	varchar(250)	NO		NULL	
telefono	varchar(15)	NO		NULL	
salario	float	NO		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> DESCRIBE Reporte;
```

Field	Type	Null	Key	Default	Extra
id_reporte	int	NO	PRI	NULL	
descripcion	varchar(250)	NO		NULL	
fecha	date	NO		NULL	
distancia_km	int	NO		NULL	
id_conductor	int	NO	MUL	NULL	
id_automovil	int	NO	MUL	NULL	

```
6 rows in set (0.00 sec)
```

Figure 5: Descripción de tablas 1

```
mysql> DESCRIBE Gastos;
```

Field	Type	Null	Key	Default	Extra
id_gastos	int	NO	PRI	NULL	
gasolina	int	NO		NULL	
fecha	date	NO		NULL	
monto	int	NO		NULL	
id_reporte	int	NO	MUL	NULL	

5 rows in set (0.00 sec)

```
mysql> DESCRIBE Documentos_automovil;
```

Field	Type	Null	Key	Default	Extra
id_doc_automovil	int	NO	PRI	NULL	
tenencia_vigencia	date	NO		NULL	
seguro_vigencia	date	NO		NULL	
verificacion_vigencia	date	NO		NULL	
tarjeta_circulacion_vigencia	date	NO		NULL	
costo	int	NO		NULL	
id_gastos	int	NO	MUL	NULL	

7 rows in set (0.00 sec)

```
mysql> DESCRIBE Mantenimiento;
```

Field	Type	Null	Key	Default	Extra
id_mantenimiento	int	NO	PRI	NULL	
descripcion	varchar(250)	NO		NULL	
fecha	date	NO		NULL	
costo	int	NO		NULL	
id_doc_automovil	int	NO	MUL	NULL	

5 rows in set (0.00 sec)

Figure 6: Descripción de tablas 2

```
mysql> DESCRIBE Documentos_conductor;
```

Field	Type	Null	Key	Default	Extra
id_doc_conductor	int	NO	PRI	NULL	
licencia_vigencia	date	NO		NULL	
seguro_medico_vigencia	date	NO		NULL	
costo	int	NO		NULL	
id_gastos	int	NO	MUL	NULL	

5 rows in set (0.00 sec)

```
mysql> DESCRIBE Danio;
```

Field	Type	Null	Key	Default	Extra
id_danio	int	NO	PRI	NULL	
costo	int	NO		NULL	
fecha	date	NO		NULL	
descripcion	varchar(250)	NO		NULL	
id_doc_automovil	int	NO	MUL	NULL	

5 rows in set (0.00 sec)

Figure 7: Descripción de tablas 3

## 5. Conclusiones

En esta práctica repasamos los conceptos básicos del diseño y manejo de bases de datos relacionales. Ya que teníamos conocimiento previo sobre el uso de llaves primarias y foráneas, la creación de tablas y la utilidad de herramientas como ERDPlus y MySQL, así que la mayor parte de la actividad fue una oportunidad para reforzar lo que ya sabíamos.

Lo único realmente nuevo para nosotros fue descubrir qué es una flotilla de autos y darnos cuenta de la importancia que tiene gestionarla de forma correcta. Entendimos que un buen control de vehículos, conductores, mantenimientos y gastos permite tomar mejores decisiones y optimizar recursos.

En general, esta práctica nos sirvió para consolidar conocimientos, repasar procesos de diseño e implementación de bases de datos, y reconocer la relevancia que un sistema bien estructurado puede tener en la administración de una flotilla.

## Referencias Bibliográficas

## References

- [1] Edenred México. (2023, 2 de octubre). *Flotilla de autos: cómo administrarla*. Edenred México.  
<https://www.edenred.mx/blog/flotilla-de-autos-como-administrarla>