

# REPORTE DE PRCTICA NO. 1

## FLOTILLA DE CARROS UBER/TAXI

ALUMNO: Cristian Cristobal Silverio  
Dr. Eduardo Cornejo-Velzquez



## 1. Introduccin

Mostrar el proceso necesario para poder realizar el diseo y, posteriormente, la implementacin de una base de datos, la cual nos ayude a llevar una administracin y control ms ptimos. Esto se logra con la ayuda de las herramientas vistas y las prximas a ser trabajadas durante nuestra formacin en la licenciatura en Ciencias Computacionales, pero con un enfoque hacia las bases de datos.

En esta prctica, usaremos el escenario de una flotilla de autos, cuyo dueo busca tener un control en la administracin de sus vehculos, los cuales ofrecen un servicio determinado.

## 2. Marco terico

En la actualidad, donde la tecnologa avanza rpidamente y se integra cada vez ms en nuestra vida cotidiana debido a las facilidades y ventajas que nos ofrece, su implementacin en tareas ms complejas, como la administracin y control de una empresa, nos proporciona numerosas herramientas que permiten reducir la complejidad de estas actividades mediante el uso de soluciones tecnolgicas especializadas en el rea en que se aplican.

Las bases de datos son una de estas herramientas, fundamentales para llevar una gestin adecuada de la informacin con la que interactuamos. Estas bases de datos nos proporcionan una gran cantidad de datos que deben ser no solo gestionados, sino tambin protegidos y procesados de manera segura, especialmente cuando se trata de datos personales, tal como lo establece el Reglamento General de Proteccin de Datos (RGPD).

### Anlisis de requerimientos

Ejemplo de cita a referencia bibliogrfica [1]

- **Gestin de Vehculos:** La base de datos debe permitir la creacin, actualizacin y eliminacin de registros de vehculos, incluyendo detalles como marca, modelo, ao, nmero de identificacin (VIN), estado de mantenimiento y asignacin de ciudad.
- **Gestin de Conductores:** La base de datos debe incluir informacin sobre los conductores, como datos personales, historial de conduccin y asignaciones de vehculos.
- **Registro de Mantenimiento:** La base de datos debe permitir el seguimiento del mantenimiento de los vehculos, registrando tanto el historial de mantenimientos realizados como las fechas de mantenimiento programadas.
- **Control de Ciudades:** La base de datos debe de incluir la informacin necesaria de las ciudades en las cuales brindaran los servicios de transporte, como seria nombre de ciudad, estado, codigo postal.
- **Control de Polizas de Seguro:** La base de datos debe de permitir la gestion y control de cada una de las polizas de seguro con las cuenta los vehculos de cada flotilla, incluyendo detalles como nmero de pliza, compaa aseguradora, fechas de inicio y vencimiento, y coberturas contratadas.
- **Control de los Servicios Ofrecidos:** La base de datos de igual manera tiene que llevar los registros de cada servicio que ofrece dicha flotilla, detalles como ciudad donde se realizo, cliente que lo solicito, costo.
- **Control de los clientes:** La base de datos debe permitir manejar la informacin de los clientes que solicita el servicio tales detalles como horario, datos personales.

### 3. Herramientas empleadas

Es una realidad que la tecnología día a día nos va ofreciendo las herramientas necesarias que nos ayudan a poder realizar tareas de una manera mas eficiente y optima. Para la solución del problema planteado hicimos uso de las herramientas necesarias para lograr nuestros objetivos, las cuales fueron:

1. **ERD Plus:** ERDPlus es una herramienta en línea gratuita utilizada para crear diagramas de entidad-relación (ERD), esquemas relacionales y esquemas estrella. Es especialmente útil para diseñar y visualizar bases de datos. Con ERDPlus, puedes:
  - Crear Diagramas ER: Representar entidades, atributos y relaciones entre ellas.
  - Generar Esquemas Relacionales: Convertir diagramas ER en esquemas relacionales que pueden ser implementados en sistemas de gestión de bases de datos (DBMS).

Cómo nos ayuda?

Facilita la comprensión de la estructura de la base de datos, nos ayuda a identificar y corregir errores en el diseño antes de la implementación.



Figure 1: Herramienta para la creación de diagramas.

2. **SQL:** SQL (Structured Query Language) es un lenguaje de consultas estándar utilizado para gestionar y manipular bases de datos relacionales. Con SQL, puedes:
  - Definir Datos: Crear y modificar la estructura de las bases de datos (tablas, índices, etc.) usando comandos como CREATE, ALTER y DROP.
  - Manipular Datos: Insertar, actualizar, eliminar y consultar datos en las bases de datos usando comandos como INSERT, UPDATE, DELETE y SELECT.
  - Controlar el Acceso a los Datos: Gestionar permisos y roles de usuarios con comandos como GRANT y REVOKE.

Cómo nos ayuda?

Gestión Eficiente de Datos: Permite realizar operaciones complejas de manera eficiente y rápida. Automatización de Tareas: Facilita la automatización de tareas repetitivas mediante scripts SQL. Integridad y Seguridad: Asegura la integridad y seguridad de los datos mediante restricciones y permisos.



Figure 2: Lenguaje de Consultas Estructurado.

3. **MySQL Server:** MySQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto, desarrollado y mantenido por Oracle Corporation. Es ampliamente utilizado en aplicaciones web y empresariales debido a su eficiencia, escalabilidad y soporte para múltiples plataformas. Con MySQL Server, puedes:

- Almacenar y Organizar Datos: Gestionar grandes volmenes de datos de manera estructurada y eficiente. - Realizar Transacciones: Soportar operaciones transaccionales que aseguran la integridad y consistencia de los datos. - Replicar Bases de Datos: Implementar replicacin para mejorar la disponibilidad y escalabilidad de las aplicaciones. Cmo nos ayuda?

- Escalabilidad: Permite manejar grandes volmenes de datos y usuarios simultneos. Fiabilidad: Ofrece alta disponibilidad y recuperacin ante fallos. - Flexibilidad: Compatible con mltiples plataformas y lenguajes de programacin, facilitando su integracin en diversos entornos.



Figure 3: Sistema Gestor de Base de Datos.

## 4. Desarrollo

### Anlisis de requisitos

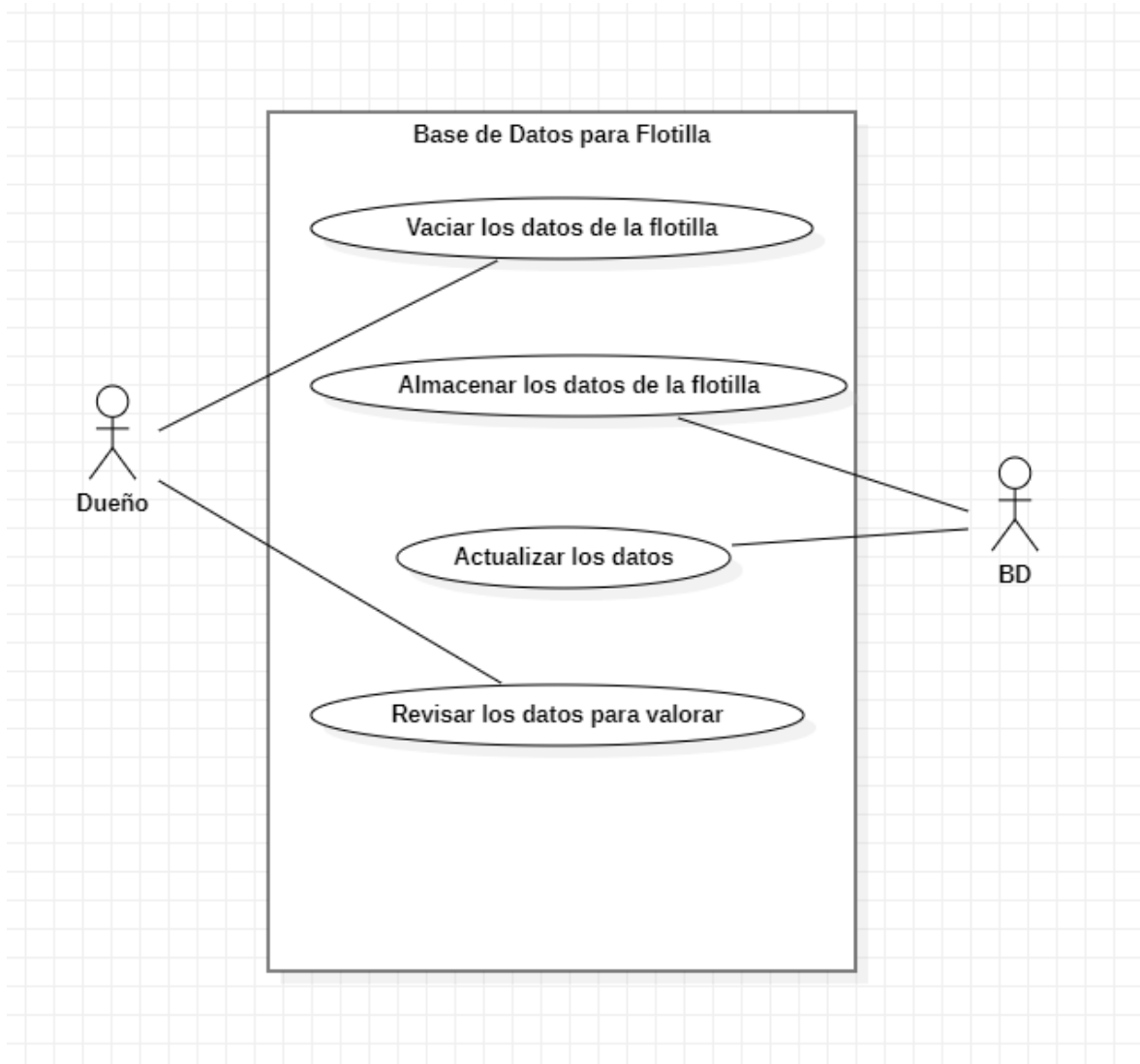


Figure 4: Diagrama que describe el funcionamiento de la base de datos.

### Modelo Entidad - Relacin

En la Tabla 1 se presenta la propuesta de Modelo Entidad - Relacin para identificar las relaciones de las entidades las cuales son los elemntos que se almacenaran en la base de datos para poder llevar un control y una organizacion optima.

En la Figura 6 se presenta la propuesta de Modelo Entidad - Relacin para la base de datos la cual nos presentan las relaciones y los atributos identificados para poder tener un analisis mas amplio.

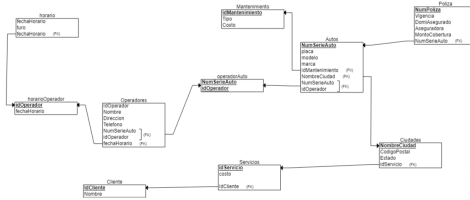


Figure 5: Modelo Relacional propuesto.

Table 1: Matriz de realaciones.

Entidades	Autos	Servicio	Ciudades	Clientes	Poliza	Operador	Horario	Mantenimiento
Autos		X	X		X	X	X	X
Servicio	X		X	X		X	X	
Ciudades	X	X		X		X		X
Clientes	X	X	X			X	X	
Poliza	X		x					
Operador	X	X	X				X	X
Horario	X	X				X		
Mantenimiento	X		X			X	X	

## Sentencias SQL

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

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

Listing 1: Crear base de datos competencia.

— *Crear la base de datos*

```
CREATE DATABASE Flotilla_Autos;
```

— *Entrar para usar la base de datos*

```
USE Flotilla_Autos;
```

— *1. Crear la tabla Cliente*

```
CREATE TABLE Cliente (  
  idCliente INT AUTOINCREMENT NOT NULL PRIMARY KEY,  
  nombre VARCHAR(30)  
);
```

— *2. Crear la tabla Servicios*

```
CREATE TABLE Servicios (  
  idServicio INT AUTOINCREMENT NOT NULL PRIMARY KEY,  
  costo INT NOT NULL,  
  idCliente INT NOT NULL,  
  FOREIGN KEY (idCliente) REFERENCES Cliente(idCliente)  
);
```

— *3. Crear la tabla Ciudades*

```
CREATE TABLE Ciudades (  
  nombreCiudad VARCHAR(30) NOT NULL PRIMARY KEY,
```

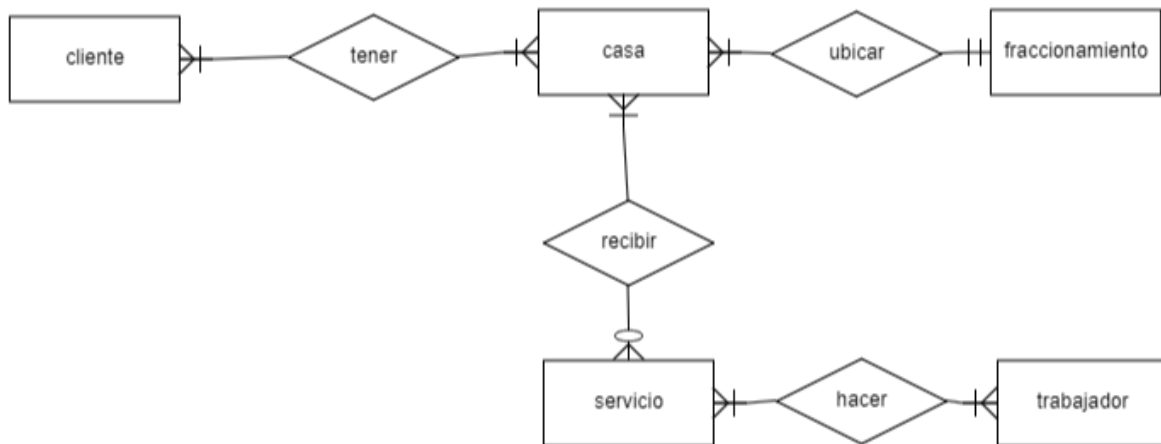


Figure 6: Modelo Entidad - Relacin propuesto.

```

codigoPostal INT NOT NULL,
idServicio INT NOT NULL,
FOREIGN KEY (idServicio) REFERENCES Servicios(idServicio)
);

```

— 4. Crear la tabla Horario

```

CREATE TABLE Horario (
fechaHorario DATE NOT NULL PRIMARY KEY,
turno VARCHAR(30),
idOperador VARCHAR(30) NOT NULL
);

```

— 5. Crear la tabla Mantenimiento

```

CREATE TABLE Mantenimiento (
idMantenimiento INT AUTOINCREMENT NOT NULL PRIMARY KEY,
tipo VARCHAR(30) NOT NULL,
costo INT NOT NULL,
fecha DATE NOT NULL,
fechaHorario DATE NOT NULL,
FOREIGN KEY (fechaHorario) REFERENCES Horario(fechaHorario)
);

```

— 6. Crear la tabla Autos

```

CREATE TABLE Autos (
numSerieAuto VARCHAR(30) NOT NULL PRIMARY KEY,
placa VARCHAR(30) NOT NULL,
modelo VARCHAR(30) NOT NULL,
idMantenimiento INT NOT NULL,
FOREIGN KEY (idMantenimiento) REFERENCES Mantenimiento(idMantenimiento),
nombreCiudad VARCHAR(30) NOT NULL,
FOREIGN KEY (nombreCiudad) REFERENCES Ciudades(nombreCiudad)
);

```

— 7. Crear la tabla Poliza



```

CREATE TABLE Poliza (
    numPoliza INT AUTO INCREMENT NOT NULL PRIMARY KEY,
    vigencia DATE NOT NULL,
    domiAsegurado VARCHAR(100) NOT NULL,
    montoCobertura FLOAT NOT NULL,
    numSerieAuto VARCHAR(30) NOT NULL,
FOREIGN KEY (numSerieAuto) REFERENCES Autos(numSerieAuto)
);

— 8. Crear la tabla Operadores
CREATE TABLE Operadores (
    idOperador INT AUTO INCREMENT NOT NULL PRIMARY KEY,
    nombre VARCHAR(30) NOT NULL,
    direccion VARCHAR(30) NOT NULL,
    telefono VARCHAR(15) NOT NULL
);

— 9. Crear la tabla horarioOperador
CREATE TABLE horarioOperador (
    idOperador INT NOT NULL,
    fechaHorario DATE NOT NULL,
FOREIGN KEY (idOperador) REFERENCES Operadores(idOperador),
FOREIGN KEY (fechaHorario) REFERENCES Horario(fechaHorario)
);

— 10. Crear la tabla operadorAuto
CREATE TABLE operadorAuto (
    numSerieAuto VARCHAR(30) NOT NULL,
    idOperador INT NOT NULL,
FOREIGN KEY (numSerieAuto) REFERENCES Autos(numSerieAuto),
FOREIGN KEY (idOperador) REFERENCES Operadores(idOperador)
);

```

Listing 2: Sentencias para poblar la base de datos.

—— *CLIENTE*

```

INSERT INTO Cliente (nombre) VALUES
( 'Magaly-Hernandez' ),
( 'Monse-Sanchez' ),
( 'Carlos-Alberto' );

```

—— *servicios*

```

INSERT INTO Servicios (costo, idCliente) VALUES
(500, 1),
(750, 2),
(300, 3);

```

—— *Ciudades*

```

INSERT INTO Ciudades (nombreCiudad, codigoPostal, idServicio) VALUES
( 'Pachuca', 1000, 1),
( 'Mineral-de-la-Reforma', 2000, 2),

```

( 'Pachuquilla' , 3000 , 3 );

— *Horario*

**INSERT INTO** Horario ( fechaHorario , turno , idOperador ) **VALUES**  
( '2024-08-22' , 'Ma ana' , 1 ),  
( '2024-08-23' , 'Tarde' , 2 ),  
( '2024-08-24' , 'Noche' , 3 );

— *Mantenimiento*

**INSERT INTO** Mantenimiento ( tipo , costo , fecha , fechaHorario ) **VALUES**  
( 'PREVENTIVO' , 100 , '2024-08-30' , '2024-08-22' ),  
( 'CORRECTIVO' , 200 , '2024-08-30' , '2024-08-23' ),  
( 'PREVENTIVO' , 150 , '2024-08-31' , '2024-08-24' );

— *Autos*

**INSERT INTO** Autos ( numSerieAuto , placa , modelo , idMantenimiento , nombreCiudad ) **VALUES**  
( 'ABC123' , 'XYZ789' , 'Toyota-TACOMA' , 1 , 'Pachuca' ),  
( 'DEF456' , 'UVW123' , 'Honda-CIVIC' , 2 , 'Pachuca' ),  
( 'GHI789' , 'RST456' , 'Mercedez-AMG' , 3 , 'Mineral-de-la-Reforma' );

— *Poliza*

**INSERT INTO** Poliza ( vigencia , domiAsegurado , montoCobertura , numSerieAuto ) **VALUES**  
( '2025-08-22' , 'PACHUCA' , 10000.0 , 'ABC123' ),  
( '2025-08-23' , 'MINERAL-DE-LA-REFORMA' , 15000.0 , 'DEF456' ),  
( '2025-08-24' , 'TODO-HIDALGO' , 20000.0 , 'GHI789' );

— *Operadores*

**INSERT INTO** Operadores ( nombre , direccion , telefono ) **VALUES**  
( 'Cristian-Cristobal' , '20-Noviembrte' , '555-1234' ),  
( 'Ana-Martinez' , 'La-raza' , '555-5678' ),  
( 'Pedro-Fernandez' , 'Tuzos' , '555-9101' );

— *horarioOperador*

**INSERT INTO** horarioOperador ( idOperador , fechaHorario ) **VALUES**  
( 1 , '2024-08-22' ),  
( 2 , '2024-08-23' ),  
( 3 , '2024-08-24' );

— *operadorAuto*

**INSERT INTO** operadorAuto ( numSerieAuto , idOperador ) **VALUES**  
( 'ABC123' , 1 ),  
( 'DEF456' , 2 ),  
( 'GHI789' , 3 );

## 5. Conclusiones

La creacin de este proyecto me favoreci enormemente en el desarrollo de mis habilidades analticas, permitindome considerar todas las caractersticas importantes para disear una base de datos lo ms escalable posible, evitando dificultades al agregar nuevas entidades o funciones.

Realizar el esquema relacional fue uno de los aspectos ms interesantes debido al grado de anlisis y planificacin necesarios para tomar decisiones correctas y mantener un ritmo adecuado, asegurando as un buen diseo y una implementacin efectiva de la solucin.

## Referencias Bibliográficas

## References

- [1] Grabowska, S.; Saniuk, S. (2022). Business models in the industry 4.0 environment results of web of science bibliometric analysis. *J. Open Innov. Technol. Mark. Complex*, 8(1), 19.