

# REPORTE DE PRÁCTICA NO. 1

Flotilla de autos Uber/taxi

ALUMNO:

Kevin Axel Chevez Quiroz



## 1. Introducción

En esta practica se realizaran los procesos para la sistematizacion de una flotilla de autos donde se llevara acabo un control, administración, mantenimiento, seguros contra accidentes y servicios de automoviles que formaran parte de esta flotilla, que utilizaran para los siguientes servicios de taxis, didi o ubers, tambien se empleara una metodologia, misma que ayudara a implementar, actualizar y visualizar todos los procesos, para que se ejecuten de manera adecuada.

Lo cual se utilizaran varios softwares para la realizacion de este sistema, tomara varias horas o dias dependiendo de la complejidad del sistema, tambien se abordaran conocimientos antes vistos durante los anteriores semestres, esto ayudara al alumno a recordar herramientas antes vistas para su construccion de esta misma

## 2. Marco teórico

El diseño de bases de datos es una herramienta fundamental para visualizar y administrar la estructura de una base de datos. Permite representar todas las entidades y sus relaciones, lo que facilita la optimización del sistema.

**2.0.1 Importancia:** Un diseño de base de datos bien organizado es vital para almacenar y recuperar información de manera eficiente y práctica.

**2.0.2 Componentes:** Abarca la definición de entidades, atributos, relaciones y restricciones, lo que permite comprender cómo se interconectan.

**2.0.3 Referencia:** La documentación oficial de MySQL y herramientas como ERDPlus son recursos útiles para el diseño y modelado de bases de datos.

### Análisis de requerimientos

En esta practica se realizara un sistema de base de datos de una flotilla de autos para que se puedan trabajar de Taxi o uber, donde se pueda establecer un control, mantenimiento y procesos que tengan. En este se podra visualizar todo el control de los autos llamadas entidades por ejemplo:

- Autos
- Operador
- Seguro
- Servicio
- Cliente
- Mantenimiento

para cada entidad sera necesario que tengan atributos para poder ejecutar ese control en el que queremos este sistema de base de datos.

1. Auto

- idauto
- Marca
- Modelo
- idOperador
- numAsientos

- idMantenimiento
- idSeguro

## 2. Operador

- idOperador
- Nombre
- Telefono

## 3. Seguro

- idSeguro
- nombre
- duracion
- costo
- idAuto

## 4. Servicio

- idServicio
- nombre
- duracion
- costo
- numPasajeros
- idAuto

## 5. Cliente

- idCliente
- nombre
- metodopago
- idAuto

## 6. Mantenimiento

- idMantenimiento
- costo
- descripcion
- idAuto

## Modelo Entidad - Relación

El modelado entidad-relación(ER) es un enfoque para diseñar bases de datos que ayuda a representar la estructura lógica de los datos. Esta técnica, propuesta por Peter Chen en 1976, es crucial en el diseño de bases de datos, ya que facilita la visualización de las entidades importantes y sus interacciones antes de que la base de datos sea implementada. En un diagrama ER, las entidades simbolizan objetos o conceptos del mundo real con existencia independiente, mientras que las relaciones muestran las conexiones entre estas entidades.

## Modelo relacional

El modelo relacional de bases de datos es un enfoque para organizar datos utilizando relaciones, a través de estructuras en forma de tabla que se componen de columnas y filas. Es el fundamento conceptual de las bases de datos relacionales y fue propuesto por Edgar F. Codd en 1969.

## SQL

SQL fue creado en la década de 1970 basándose en el modelo de datos relacional. Originalmente se conocía como SEQUEL, que en inglés significa "Structured English Query Language". Posteriormente, el nombre fue abreviado a SQL. Oracle, anteriormente llamado Relational Software, fue la primera empresa en ofrecer un sistema comercial de gestión de bases de datos relacionales basado en SQL.

## 3. Herramientas empleadas

**ERD Plus** es una herramienta en línea que se utiliza para crear diagramas de entidad-relación (ERD), fundamentales en el diseño de bases de datos. Esta herramienta ofrece una interfaz intuitiva que facilita a los usuarios el diseño gráfico de modelos de bases de datos de manera sencilla.

**Modelado de Entidad-Relación:** ERDPlus permite a los usuarios crear modelos de entidad-relación a través de la representación gráfica de entidades, atributos, relaciones y cardinalidades. Esto ayuda a visualizar cómo las diferentes entidades dentro de una base de datos están interconectadas.

**MySQL Server** es uno de los sistemas de gestión de bases de datos relacionales (RDBMS) más reconocidos y utilizados en todo el mundo. MySQL es conocido por ser un software de código abierto, lo que significa que está disponible de manera gratuita y es ampliamente empleado en proyectos de distintos tamaños, desde aplicaciones web hasta sistemas empresariales, y más.

## 4. Desarrollo

### Análisis de requisitos

Describir los requisitos principales del caso de estudio. Se sugiere utilizar alguna herramienta de ingeniería de software para presentar los requisitos, por ejemplo diagramas UML.

### Modelo Entidad - Relación

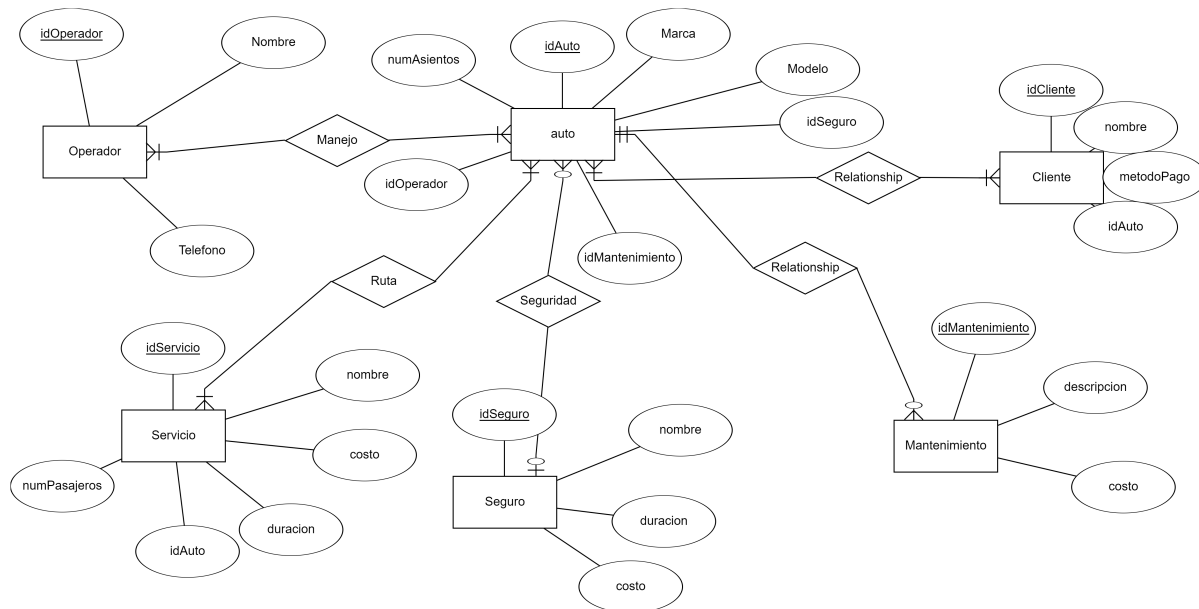


Figure 1: Modelo Entidad - Relación propuesto.

En el desarrollo de una base de datos para la gestión de una flota de autos, es fundamental realizar un análisis detallado de los requisitos. Este análisis permitirá identificar las necesidades del sistema, las funciones que deben implementarse y las entidades clave que componen el modelo de datos. A continuación, se describen los principales requisitos del caso de estudio, apoyados por diagramas UML.

#### 4.2.1 Requisitos Funcionales

Los requisitos funcionales especifican las funciones y operaciones que el sistema debe ser capaz de realizar:

1. **Gestión de Autos:** El sistema debe permitir la creación, actualización, eliminación y consulta de registros de autos. Cada auto debe estar vinculado a un seguro y su estado (activo, en mantenimiento, dado de baja) debe ser rastreable.
2. **Gestión de Operador:** El sistema debe manejar la información de los operadores, incluyendo datos personales, licencia y asignación de autos. Debe ser posible consultar qué chofer está asignado a un auto en un momento determinado.
3. **Gestión de Servicios:** El sistema debe registrar los servicios prestados, incluyendo detalles como el chofer, el auto, la ruta (origen y destino), la fecha y el costo del servicio. También debe ser posible consultar el historial de servicios realizados por un chofer o un auto específico.
4. **Gestión de Clientes:** El sistema debe permitir la administración de los datos de los clientes, como nombre, contacto y servicios contratados. Debe ser posible consultar el historial de servicios contratados por cada cliente.

5. **Gestión de Mantenimiento:** El sistema debe registrar las actividades de mantenimiento realizadas en los autos, incluyendo el tipo de mantenimiento, la fecha y la descripción de las tareas realizadas. Debe ser posible consultar el historial de mantenimiento de cada auto.
6. **Gestión de Seguros:** El sistema debe gestionar la información de los seguros asociados a los autos, incluyendo la compañía aseguradora, tipo de cobertura y vigencia.

#### 4.2.2 Requisitos No Funcionales

Los requisitos no funcionales definen los atributos de calidad del sistema, como usabilidad, rendimiento y seguridad:

1. **Escalabilidad:** El sistema debe ser capaz de manejar un aumento en la cantidad de autos, choferes, clientes y servicios sin afectar el rendimiento.
2. **Seguridad:** El acceso al sistema debe estar restringido mediante autenticación de usuarios, permitiendo que solo usuarios autorizados realicen operaciones de creación, lectura, actualización y eliminación (CRUD). Los datos sensibles, como la información de contacto de choferes y clientes, deben protegerse mediante cifrado.
3. **Disponibilidad:** El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad mínimo, dado que la operación de la flotilla de autos podría ser continua.
4. **Usabilidad:** La interfaz del sistema debe ser intuitiva y fácil de usar, permitiendo que los usuarios realicen las operaciones necesarias sin requerir un entrenamiento extenso.
5. **Integridad de los Datos:** El sistema debe asegurar la consistencia e integridad de los datos mediante la implementación de restricciones, validaciones y el uso de transacciones para operaciones críticas.

#### 4.2.3 Diagramas UML

Para representar los requisitos identificados, se pueden utilizar diagramas UML (Lenguaje Unificado de Modelado), que son herramientas estándar en la ingeniería de software para visualizar, especificar, construir y documentar los componentes de un sistema.

**Diagrama de Casos de Uso** Un diagrama de casos de uso muestra las interacciones entre los actores (usuarios u otros sistemas) y las funcionalidades del sistema.

- **Actores:**

- Administrador: Usuario responsable de la gestión de autos, choferes, seguros, clientes y mantenimiento.
- Chofer: Usuario que puede consultar sus servicios asignados y el estado del auto.
- Cliente: Usuario que puede consultar el historial de servicios contratados.

- **Casos de Uso:**

- Gestionar Autos
- Gestionar Operador
- Gestionar Servicios
- Gestionar Clientes
- Gestionar Mantenimiento
- Gestionar Seguros
- Consultar Historial de Servicios
- Consultar Estado de Auto

**Diagrama de Clases** Un diagrama de clases muestra la estructura estática del sistema, definiendo las clases (entidades) que forman parte del sistema y las relaciones entre ellas.

Estas representaciones visuales proporcionan una base clara para el diseño e implementación del sistema, asegurando que todos los requisitos identificados se aborden durante el desarrollo. Además, facilitan la comunicación entre los desarrolladores y otros interesados en el proyecto.

Table 1: Matriz de realaciones.

Entidades	Auto	Operador	Seguro	Servicio	Cliente	Mantenimiento
Auto		X	X	X	X	X
Operador	X			X	X	X
Seguro	X					
Servicio	X					
cliente	X	X		X		
Mantenimiento	X		X			

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

## Modelo relacional

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

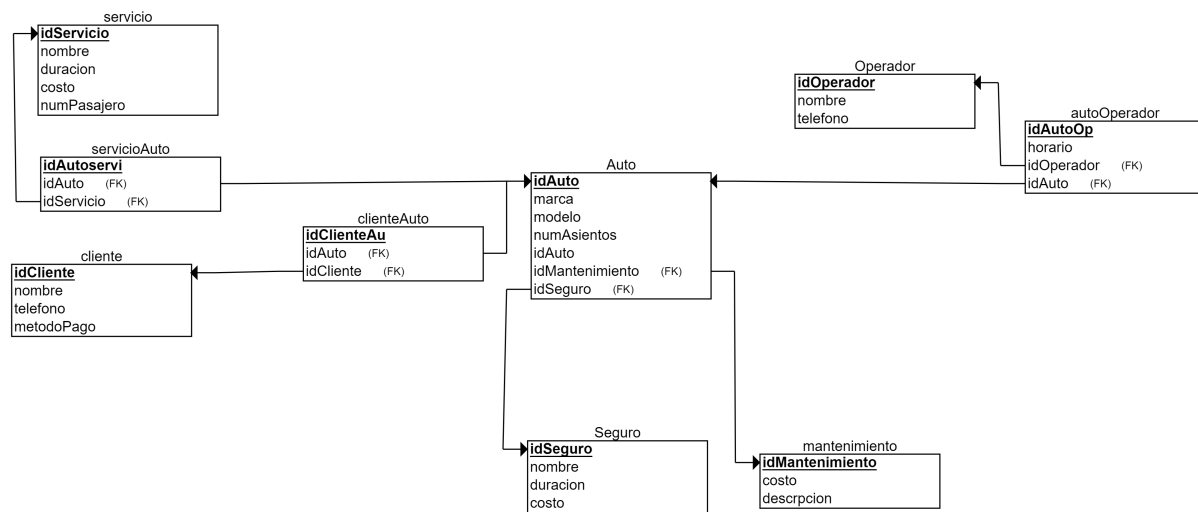


Figure 2: Modelo Relacional propuesto.

## Sentencias SQL

create database Flotilla;

– Tabla: Auto –

```
CREATE TABLE Auto ( idAuto INT PRIMARY KEY AUTO-INCREMENT, Marca VARCHAR(50) NOT NULL, Modelo VARCHAR(50) NOT NULL, idOperador int NOT NULL numAsientos int NOT NULL idMantenimiento int NOT NULL idSeguro int NOT NUL FOREIGN KEY (idOperador) REFERENCES operador(idOperador), FOREIGN KEY (idMantenimiento) REFERENCES Mantenimiento(idMantenimiento), FOREIGN KEY (idSeguro) REFERENCES seguro(idSeguro) );
```

– Tabla: Operador –

```
CREATE TABLE Operador ( idOoperador INT PRIMARY KEY AUTO-INCREMENT, Nombre VARCHAR(100) NOT NULL, Telefono VARCHAR(15) NOT NULL );
```

– Tabla: Cliente –

```
CREATE TABLE Cliente ( idCliente INT PRIMARY KEY AUTO-INCREMENT, Nombre VARCHAR(100) NOT NULL, metodoPago VARCHAR(50), idAuto int NOT NUL FOREIGN KEY (idAuto) REFERENCES auto(idAuto) );
```

– Tabla: Servicio –

```
CREATE TABLE Servicio ( idServicio INT PRIMARY KEY AUTO-INCREMENT, Nombre VARCHAR(50), Duracion VARCHAR(50), Costo VARCHAR(50), numPasajeros int, idAuto int NOT NUL FOREIGN KEY (idAuto) REFERENCES auto(idAuto) );
```

– Tabla: Mantenimiento –

```
CREATE TABLE Mantenimiento ( idMantenimiento INT PRIMARY KEY AUTO-INCREMENT, Descripcion TEXT, Costo VARCHAR(10), idAuto int NOT NUL FOREIGN KEY (idAuto) REFERENCES auto(idAuto) );
```

– Tabla: Seguro –

```
CREATE TABLE Seguros ( idSeguro INT PRIMARY KEY AUTO-INCREMENT, idAuto INT, Nombre
```



VARCHAR(100) NOT NULL, Duracion varchar(60), costo varchar(50), FOREIGN KEY (AutoID) REFERENCES Autos(AutoID) );

INSERT INTO Auto (Marca, Modelo,idOperador, numAsiento, idMantenimiento, idSeguro) VALUES ('Chevrolet','2005','1','4','1','1'), ('Mazda','2024','2','4','2','2'), ('Audi','2024','3','4','3','3'), ('Mercedes','2018','4','6','4','4');

INSERT INTO Operador (nomnbre, telefono) VALUES ('Axel','5587344046'), ('Itzel','554181616'), ('Fernando','569798941'), ('Jesus','7787482816');

INSERT INTO Cliente (nomnbre, metodoPago, idAuto) VALUES ('Marian','Efectivo',1), ('Miguel','Credito',2), ('Saul','Transferencia',3), ('Damaris','Transferencia',4);

INSERT INTO Servicio (Nombre, Duracion, Costo, numPasajeros,idAuto) VALUES ('Premium','1 hora','60','6','4'), ('Platinum','1 hora','120','6','4'), ('Estandar','30 minutos','10','4','3');

INSERT INTO Mantenimiento (decripcion, costo, idAuto) VALUES ('Cambio de valatas','400','1'), ('Cambiodeaceite','260','2'), ('Cambio de filtros','100','1'), ('Cambio de bujias','600','4');

INSERT INTO Seguro (idAuto, Nombre, duracion, costo) VALUES ('1','seguros GNP','4 años','20000'), ('2','seguros Ganzo','1 años','10000'), ('3','seguros Bancomer','3 años','15000'), ('4','seguros Garza','10 años','50000');

## 5. Conclusiones

En conclusion esta practica nos ayudo a reconocer y recordar lo que se aplico semestres anteriores ya que ayudo a recuperar un conocimiento que estaba guardado, tambien ayudo a ver problemas de sintaxis que teniamos por lo tanto se concluye que la logistica de la flotilla de auto puede funcionar de manera adecuada.

## 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.
- [Godaddy] alvo, L. (2024, 23 abril). *Guía sobre cómo crear una base de datos*. GoDaddy Resources - Spain. <https://www.godaddy.com/resources/es/email/crear-base-de-datos>