

# REPORTE DE PRÁCTICA NO. 1

**NOMBRE DE LA PRÁCTICA:** Flotilla de autos Taxi/Uber

**ALUMNO:** Romo Guzmán Jesús

**Dr. Eduardo Cornejo-Velázquez**



## 1. Introducción

En la actualidad, la gestión eficiente de una flotilla de vehículos es crucial para empresas que ofrecen servicios de transporte, como las compañías de taxis. La creciente demanda de servicios de movilidad y la necesidad de ofrecer un servicio seguro y confiable, requieren de una infraestructura tecnológica robusta y escalable que permita la administración efectiva de los recursos.

Este proyecto se enfoca en el diseño e implementación de una base de datos distribuida para una flotilla de autos que brinda un servicio de taxi. El objetivo principal es desarrollar un sistema que garantice el control, escalabilidad y consistencia de los datos en un entorno distribuido, permitiendo la administración de los autos, operadores, servicios y mantenimiento de los autos.

La base de datos distribuida está diseñada para soportar grandes volúmenes de datos generados por la operación diaria de la flotilla, incluyendo la hora y lugar donde fue hecho el servicio, el historial de viajes, la información de los operadores, las interacciones con los clientes y la información de los autos. El sistema permitirá que la empresa tenga una mejor administración de la flotilla de autos, la cual ofrece un servicio de taxi.

Además, la solución propuesta debe ser capaz de manejar la tolerancia a fallos, asegurando que la operación continúe sin interrupciones, incluso en caso de fallos en algunos de los nodos del sistema. Para ello, se evaluarán y seleccionarán tecnologías de bases de datos distribuidas que permitan cumplir con estos requerimientos, balanceando aspectos críticos como la consistencia, la disponibilidad y la partición de los datos.

## 2. Marco teórico

Esta base de datos distribuida es para la administración de una flotilla de autos que brindan un servicio de taxi, abarca los conceptos fundamentales y teorías que sustentan el diseño, implementación, y operación de dicha base de datos. Este marco teórico sirve para contextualizar el proyecto y justificar las decisiones técnicas tomadas.

**Conceptos Básicos:** Una base de datos relacional es un sistema que organiza los datos en tablas estructuradas que se relacionan entre sí mediante claves primarias y foráneas. Cada tabla representa una entidad y contiene atributos (columnas) que describen las características de esa entidad. Las claves primarias aseguran la unicidad de cada registro dentro de una tabla, mientras que las claves foráneas establecen las relaciones entre tablas, permitiendo la integridad referencial.

**Modelo Entidad-Relación (ER):** El modelo Entidad-Relación (ER) es un modelo conceptual crucial para el diseño de bases de datos. En este modelo, las entidades (por ejemplo, **Auto**, **Cliente**, **Servicio**) se representan como tablas, y las relaciones entre estas entidades se modelan utilizando claves foráneas. El modelo ER facilita la visualización de cómo se interconectan los datos, lo que ayuda a planificar y estructurar la base de datos de manera eficiente. Este modelo permite identificar claramente las entidades principales y sus interacciones, simplificando el diseño y la implementación del esquema de la base de datos.

## 3. Herramientas empleadas

### 3.0.1 ERDPlus

ERDPlus es una herramienta en línea utilizada para crear diagramas de modelos entidad-relación (ER) y esquemas relacionales. Para este proyecto, ERDPlus se utilizó en las siguientes etapas:

#### 1. Diseño Conceptual del Modelo Entidad-Relación (ER):

- Se utilizó ERDPlus para modelar las entidades clave del sistema (**Auto**, **Cliente**, **Servicio**, **Operador**, etc.) y sus relaciones.
- Esta herramienta permitió identificar visualmente las entidades y cómo están relacionadas entre sí, asegurando que todas las relaciones relevantes, como 1 y Muchos, estuvieran correctamente representadas.
- El diagrama ER crea sirvió como un mapa conceptual que guio el proceso de normalización y la creación de las tablas en la base de datos.

#### 2. Generación del Modelo Relacional:

- Después de definir el modelo ER, ERDPlus facilitó la conversión de este modelo conceptual en un esquema relacional, que incluye la definición de las tablas, sus atributos, claves primarias y foráneas.
- Esto permitió visualizar cómo las entidades y sus relaciones se transformarían en tablas de base de datos, lo cual fue esencial para la implementación en MySQL Server.

### 3.0.2 MySQL Workbench

**MySQL Workbench** es una herramienta visual de diseño y administración de bases de datos para MySQL. **Generación de Sentencias SQL:**

- **Generación Automática de Sentencias:** MySQL Workbench facilitó la generación automática de las sentencias `CREATE TABLE` necesarias para implementar el esquema relacional definido. Estas sentencias SQL incluían la definición de todas las tablas, claves primarias, claves foráneas, y otros constraints (restricciones) según lo especificado en el modelo.
- **Modificación Manual y Personalización:** Aunque MySQL Workbench puede generar automáticamente las sentencias SQL, también permitió personalizarlas y modificarlas manualmente para ajustar detalles específicos antes de ejecutar las operaciones de creación en la base de datos.

itemize

## 4. Desarrollo

### Análisis de requisitos

Estos requisitos definen las funcionalidades específicas que la base de datos debe soportar para cumplir con los objetivos del sistema.

- **Gestión de Flotilla de Autos:**

- Registrar y mantener información detallada sobre cada vehículo, incluyendo marca, modelo, año, placa, seguro asociado, y ciudad de operación.
- Registrar y gestionar el historial de mantenimientos de cada vehículo, con detalles como fecha, tipo de mantenimiento, costo, operador responsable, y ciudad.

- **Gestión de Servicios:**

- Registrar cada servicio de taxi realizado, incluyendo detalles como fecha, hora de inicio y fin, cliente, operador, auto utilizado, y ciudad.
- Asignar autos y operadores a servicios específicos en función de la disponibilidad y el horario.

- **Gestión de Clientes:**

- Registrar y mantener información de los clientes, como nombre, contacto, dirección, y ciudad.
- Permitir el seguimiento de la frecuencia de uso y el historial de servicios prestados a cada cliente.

- **Gestión de Operadores:**

- Registrar información sobre los operadores (conductores), incluyendo nombre, licencia, contacto, y ciudad de operación.
- Gestionar los horarios de los operadores y su asignación a los autos y servicios.

- **Gestión de Seguros:**

- Registrar y gestionar la información del seguro asociado a cada auto, incluyendo la compañía aseguradora, número de póliza, y fecha de vencimiento.
- Relacionar la información del seguro con el auto y la ciudad correspondiente.

- **Planificación y Gestión de Horarios:**

- Registrar y gestionar los horarios de operación, tanto para los servicios de taxi como para los operadores y mantenimiento de los vehículos.
- Garantizar la correcta asignación de autos y operadores en función de la disponibilidad y horarios.

- **Consultas y Reportes:**

- Generar reportes y realizar consultas sobre el estado de la flotilla, historial de servicios, desempeño de operadores, y uso de clientes.
- Consultar la disponibilidad de autos y operadores en tiempo real para la asignación de servicios.

### Modelo Entidad - Relación

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

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

### Modelo relacional

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

Table 1: Matriz de realaciones.

Entidades	Auto	Servicio	Ciudad	Cliente	Seguro	Operador	Mantenimiento	Horario
Auto		X	X	X	X	X	X	X
Servicio	X		X	X		X		X
Ciudad	X	X		X	X	X	X	
Cliente	X	X	X			X		X
Seguro	X		X					
Operador	X	X	X				X	X
Mantenimiento	X		X			X		X
Horario	X	X		X		X	X	

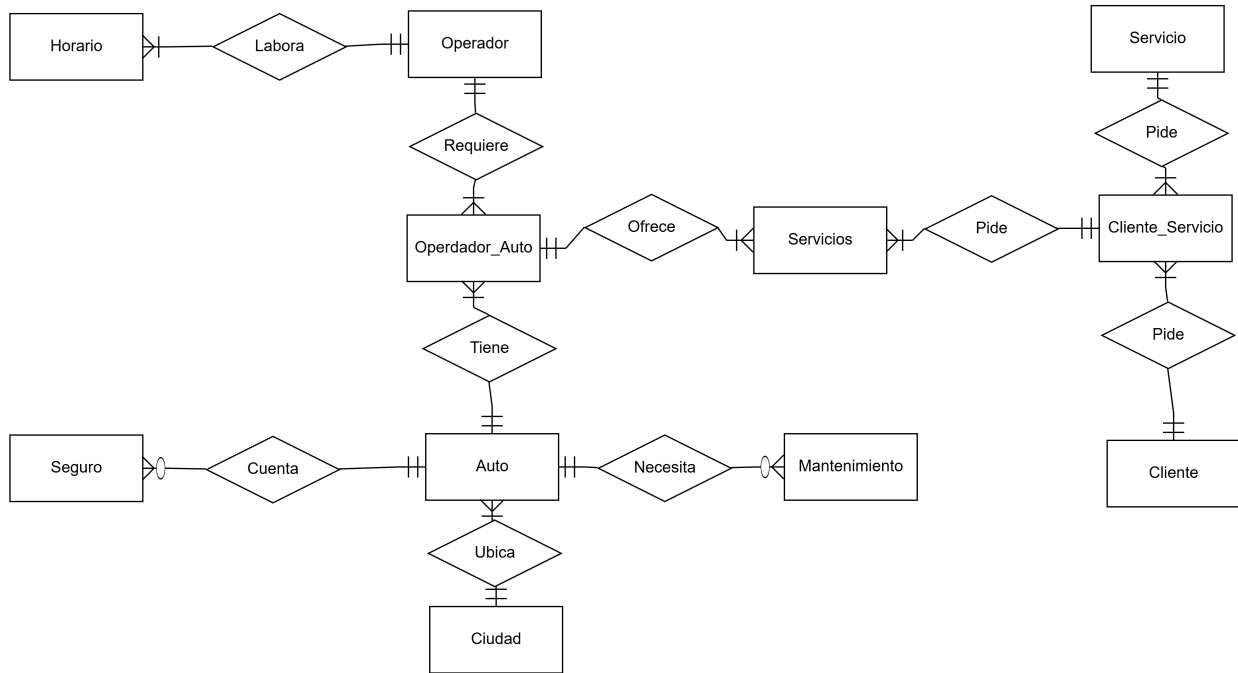


Figure 1: Modelo Entidad - Relación propuesto.

## Sentencias SQL

Presentar las sentencias para crear la base de datos y tablas.

- Crear la base de datos `CREATE DATABASE FlotillaTaxisBD; USE FlotillaTaxisBD;`
- Crear la tabla Ciudad `CREATE TABLE Ciudad ( ID_Ciudad INT NOT NULL AUTO_INCREMENT, Nombre VARCHAR(50), PRIMARY KEY (ID_Ciudad));`
- Crear la tabla Seguro `CREATE TABLE Seguro ( ID_seguro INT AUTO_INCREMENT PRIMARY KEY, Nombre_Aseguro VARCHAR(50), PRIMARY KEY (ID_seguro));`
- Crear la tabla Auto `CREATE TABLE Auto ( ID_Auto INT AUTO_INCREMENT PRIMARY KEY, Número_serie VARCHAR(50), PRIMARY KEY (ID_Auto));`
- Crear la tabla Cliente `CREATE TABLE Cliente ( ID_cliente INT AUTO_INCREMENT PRIMARY KEY, Nombre VARCHAR(50), PRIMARY KEY (ID_cliente));`
- Crear la tabla Operador `CREATE TABLE Operador ( ID_operador INT AUTO_INCREMENT PRIMARY KEY, Nombre VARCHAR(50), PRIMARY KEY (ID_operador));`
- Crear la tabla Mantenimiento `CREATE TABLE Mantenimiento ( ID_Mantenimiento INT AUTO_INCREMENT PRIMARY KEY, Fecha DATE, PRIMARY KEY (ID_Mantenimiento));`
- Crear la tabla Horario `CREATE TABLE Horario ( ID_Horario INT AUTO_INCREMENT PRIMARY KEY, Dia ENUM('L', 'M', 'X', 'J', 'V', 'S'), PRIMARY KEY (ID_Horario));`
- Crear la tabla Servicio `CREATE TABLE Servicio ( ID_servicio INT AUTO_INCREMENT PRIMARY KEY, Fecha DATE, PRIMARY KEY (ID_servicio));`
- Crear la tabla Auto\_cliente `CREATE TABLE Servicio_cliente (ID_servicio INT, ID_cliente INT, PRIMARY KEY (ID_servicio, ID_cliente));`
- Crear la tabla Operador\_Auto `CREATE TABLE Operador_Auto (ID_Auto INT, ID_operador INT, PRIMARY KEY (ID_Auto, ID_operador));`
- Crear la tabla Operador\_Auto\_cliente\_servicio `CREATE TABLE Operador_Auto_cliente_servicio (ID_operador INT, ID_Auto INT, ID_servicio INT, PRIMARY KEY (ID_operador, ID_Auto, ID_servicio));`

Listing 1: Crear base de datos competencia.

```
CREATE DATABASE competencia .
```

## 5. Conclusiones

En conclusión, en esta práctica, aprendí y recordé en como diseñar una base de datos para la administración de una flotilla de autos, donde tuve muchos problemas para diseñar cada tabla y también en como conectar cada tabla.



## 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.

