

REPORTE DE PRÁCTICA NO. 1

Flotilla de Autos

ALUMNO: Braulio Adrián Pérez Orta
Dr. Eduardo Cornejo-Velázquez



1. Introducción

Incluir una descripción de los que trata la práctica.

En un entorno dinámico de transporte pagado, la administración eficiente de flotas de vehículos es esencial para garantizar la calidad del servicio y la satisfacción del cliente. Los objetivos de diseño de la base de datos distribuida se centrarán en lo siguiente:

Autos:

registre información detallada sobre cada vehículo en la flota. Esto debería incluir su modelo, año de fabricación, kilometraje, estado de mantenimiento y datos específicos, como la capacidad de pasajeros o su tipo de combustible.

Conductores:

administre los perfiles de los conductores, incluida su licencia, antecedentes, horario de trabajo y calificaciones. La importancia de asignar al conductor al vehículo correspondiente es fundamental para la operación.

Servicios:

La base de datos almacenará información sobre los servicios prestados por los vehículos. Esto incluye detalles de los viajes, tarifas, rutas y tiempos de inicio y finalización.

Clientes

Registrar datos de los clientes, como nombres, direcciones, preferencias y transacciones. Esto permitirá un seguimiento personalizado y una comunicación efectiva.

Mantenimiento:

Llevar un registro de los mantenimientos programados y correctivos para cada vehículo. Esto garantiza la seguridad y prolonga la vida útil de la flota.

Seguros:

Almacenar información sobre las pólizas de seguro de los vehículos y los conductores. Esto es fundamental para la gestión de riesgos y la protección financiera.

2. Marco teórico

// Para construir el marco teórico deben consultar bibliografía y citarla de forma correcta de acuerdo con el ejemplo //

Análisis de requerimientos

Ejemplo de cita a referencia bibliográfica [1] para incorporarla al documento. Se les permite registrar información sobre sus clientes, obteniendo su nombre, dirección, número de teléfono y método de pago. Cada cliente registrado se puede mapear con los viajes que realiza. Además, redactan detalles de las pólizas de seguro para cada vehículo, incluidas las fechas de vencimiento y otras coberturas. Otra herramienta les ofrece registrar los servicios de mantenimiento que se le realizan a sus autos; en ella se registran fechas, descripciones de los servicios y montos gastados. También se les permite almacenar información sobre cada vehículo, tal como su modelo, año, placa y tipo, por ejemplo, sedán o SUV. Para ellos se les permite asentar los autos con los conductores a los que estos pertenecen, y asociar los con su seguro. Por último, se les permite registrar datos de los conductores, tales como su nombre licencia de conducir o contacto, entre otros; y asociar cada conductor con los autos que maneja, así como llevar un seguimiento completo de cada viaje realizado asegurando cada pasaje y la seguridad de tanto conductor como clientela.

Modelo Entidad - Relación

Modelo Entidad-Relación : es una técnica gráfica utilizada para expresar la estructura conceptual de la base de datos. Se basa en la premisa de que la base de datos está hecha de entidades y relaciones entre ellas. Los elementos fundamentales del MER son sus entidades y sus propiedades que se presentan en diagramas.

Modelo relacional

Tablas (Relaciones): Imagina las tablas como hojas de cálculo. Cada tabla representa un conjunto de datos (por ejemplo, “Clientes” o “Productos”). Las filas en la tabla son las tuplas (registros), y las columnas son los atributos (como “Nombre” o “Precio”). Claves Primarias y Foráneas: Las claves primarias son identificadores únicos para cada fila en una tabla. Por ejemplo, el número de identificación de un cliente. Las claves foráneas establecen relaciones entre tablas. Por ejemplo, si una tabla “Pedidos” tiene una columna “ID_Cliente” que se relaciona con la tabla “Clientes”. SQL (Structured Query Language): Es el lenguaje que usamos para hacer preguntas y modificar datos en estas bases de datos. Con SQL, puedes buscar información, agregar registros, actualizar valores y más.

SQL

lenguaje estándar utilizado para interactuar con bases de datos relacionales. Permite realizar consultas (SELECT), inserciones (INSERT), actualizaciones (UPDATE) y eliminaciones (DELETE) de datos. entre otras operaciones básicas de el lenguaje.

3. Herramientas empleadas

Describir qué herramientas se han utilizado...

1. ERD Plus. Describir cuál es su tipo y para qué se utiliza. Herramienta de modelado gráfico para diseñar esquemas de bases de datos relacionales. ERDPlus permite crear diagramas entidad-relación (DER) de manera intuitiva a través de su interfaz en línea. Presenta una solución completa tanto para usuarios aficionados como profesionales de la ingeniería de datos.

Tipo:

Basado en la nube: El modelo de servicio web hace que ERDPlus sea accesible desde cualquier dispositivo con solo utilizar un navegador. No se requiere descarga ni instalación local.

Libre y pago: Existe una versión gratuita con funciones básicas. Asimismo, las opciones de plan de pago añaden características avanzadas para proyectos más complejos.

Funcionalidad:

Diagramación de entidades y relaciones: Permite visualizar y modelar la estructura conceptual de una base de datos a través de diagramas entidad-relación. Estos diagraman las tablas, atributos y vínculos entre ellas.

Diseño lógico: Junto a los DER, genera esquemas lógicos de la base de datos. Estos especifican las tablas, columnas, claves primarias y foráneas que conforman el diseño final.

2. MySQL Command Line. línea de comandos de MySQL es una herramienta esencial para interactuar con bases de datos MySQL directamente desde la terminal o el símbolo del sistema.

Algunos comandos fundamentales para administrar bases de datos y tablas son: `SHOW DATABASES;;` Muestra las bases de datos disponibles. `USE database_name;;` Selecciona una base de datos específica. `SHOW TABLES;;` Muestra las tablas en la base de datos actual. `DESCRIBE table_name;` o `SHOW COLUMNS FROM table_name;;` Muestra la estructura de una tabla. `SELECT * FROM table_name;;` Obtiene todos los registros de una tabla. todo esto te da la posibilidad de Trabajar con datos.

4. Desarrollo

Análisis de requisitos

\item Seguimiento de los vehículos de la flotilla. \item Gestión de los operadores responsables de los vehículos. \item Registro y control de los servicios ofrecidos y solicitados. \item Mantenimiento de un historial detallado de todos los trabajos de mantenimiento realizados en los vehículos. \item Administración de los contratos de seguros para los vehículos. \item Gestión de clientes que reciben los servicios de transporte
Todo esto se menciona con anterioridad en la Introduccion ya que son parte del enfoque total de el proyecto.por lo cual debe permitir eliminacion, actualización y escritura de datos de cada uno de los tipos de datos antes descritos.

Modelo Entidad - Relación

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

Table 1: Matriz de realaciones.

Entidades	auto	operador	cliente	servicio	mantenimiento	seguro
auto	X	X	X		X	X
operador	X	X		X		
cliente	X		X	X		
servicio		X	X	X		
mantenimiento	X				X	
seguro	X					X

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

Modelo relacional

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

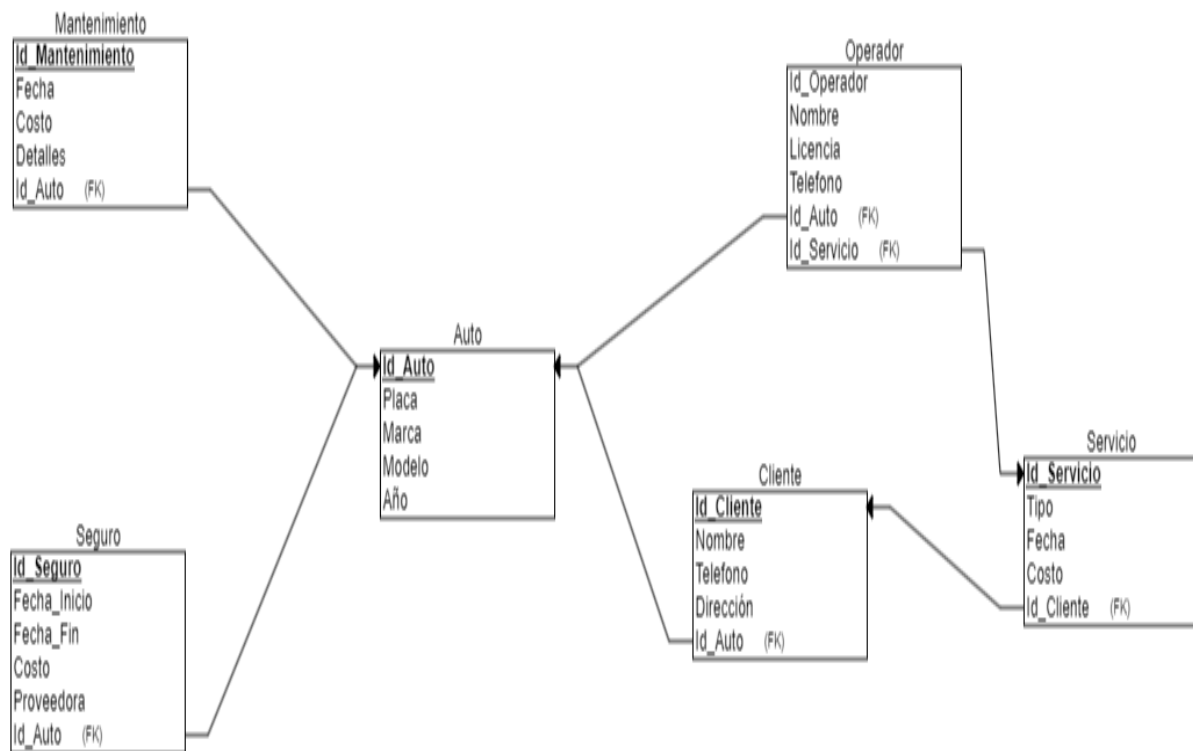


Figure 1:

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 competencia.

Listing 1: Crear base de datos competencia.

```
CREATE DATABASE Flotilla
USE Flotilla;

CREATE TABLE Auto
(Id\_Auto INT NOT NULL AUTO\INCREMENT PRIMARY KEY,
Placa VARCHAR (10), Marca VARCHAR (30), Modelo VARCHAR(50),
Anho INT NOT NULL );

CREATE TABLE Seguro
(id\_Seguro INT NOT NULL AUTO\INCREMENT
PRIMARY KEY, Fecha\_inicio VARCHAR (20), Fecha\_fin
VARCHAR(20),
Costo INT NOT NULL, Id\_Auto INT, Proveedor VARCHAR (50),
FOREIGN KEY (Id\_Auto) REFERENCES Auto (Id\_Auto));

CREATE TABLE Mantenimiento
(id\_Mantenimiento INT NOT NULL AUTO\INCREMENT
PRIMARY KEY,
Fecha VARCHAR (20), Detalles VARCHAR(400), Costo INT
NOT NULL, Id\_Auto INT, FOREIGN KEY (Id\_Auto)
REFERENCES Auto (Id\_Auto));

CREATE TABLE Cliente
(Id\_Cliente INT NOT NULL AUTO\INCREMENT PRIMARY KEY,
Nombre VARCHAR(70), Direccion VARCHAR (90), Telefono INT,
Id\_Auto INT, FOREIGN KEY (Id\_Auto) REFERENCES Auto (Id\_Auto));

CREATE TABLE Servicio
(id\_Servicio INT NOT NULL AUTO\INCREMENT PRIMARY KEY,
Fecha VARCHAR (20), Tipo VARCHAR(50), Costo INT NOT NULL,
Id\_Cliente INT, FOREIGN KEY (Id\_Cliente)
REFERENCES Cliente (Id\_Cliente));

CREATE TABLE Operador
(id\_Operador INT NOT NULL AUTO\INCREMENT
PRIMARY KEY, Nombre VARCHAR (90), Licencia VARCHAR(40)
, Telefono INT NOT NULL, Id\_Auto INT, Id\_Servicio INT,
FOREIGN KEY (Id\_Auto) REFERENCES Auto (Id\_Auto),
FOREIGN KEY (Id\_Servicio) REFERENCES Servicio (Id\_Servicio));
```


INSERT INTO Cliente (nombre,Direccion ,Telefono) **VALUES**

('Juan-Perez ', 'Calle-Principal-123 ', '555123457 ');
Query OK, 1 row affected (0.00 sec)

INSERT INTO Cliente (nombre,Direccion ,Telefono)

VALUES ('Maria-Rodriguez ', 'Calle-del-Sol-789 ', '555987654 ')
;
Query OK, 1 row affected (0.00 sec)

INSERT INTO Cliente (nombre,Direccion ,Telefono)

VALUES ('Pedro-Garcia ', 'Avenida-Central-567 ', '555555555 ');
Query OK, 1 row affected (0.00 sec)

INSERT INTO Cliente (nombre,Direccion ,Telefono)

VALUES ('Ana-Ramirez ', 'Plaza-Mayor-321 ', '555222333 ');
Query OK, 1 row affected (0.00 sec)

INSERT INTO Cliente (nombre,Direccion ,Telefono)

VALUES ('Luis-Martinez ', 'Boulevard-Norte-567 ', '555444777 ');
Query OK, 1 row affected (0.00 sec)

INSERT INTO Cliente (nombre,Direccion ,Telefono)

VALUES ('Laura-Sanchez ', 'Callejon-Sur-890 ', '555888999 ');
Query OK, 1 row affected (0.00 sec)

INSERT INTO Cliente (nombre,Direccion ,Telefono)

VALUES ('Carlos-Torres ', 'Avenida-Oeste-234 ', '555666111 ');
Query OK, 1 row affected (0.00 sec)

INSERT INTO Cliente (nombre,Direccion ,Telefono)

VALUES ('Isabel-Lopez ', 'Calle-Este-678 ', '555333222 ')

INSERT INTO Seguro (Fecha_inicio , Fecha_fin , Costo , Id_Auto , Proveedor)
VALUES

('2024-08-01 ', '2025-08-01 ', 500, 1, 'Seguros-ABC'),
('2024-07-15 ', '2025-07-15 ', 450, 2, 'Seguros-XYZ'),
('2024-06-20 ', '2025-06-20 ', 400, 3, 'Seguros-DEF'),
('2024-05-10 ', '2025-05-10 ', 550, 4, 'Seguros-GHI'),
('2024-04-05 ', '2025-04-05 ', 600, 5, 'Seguros-JKL'),
('2024-03-01 ', '2025-03-01 ', 520, 6, 'Seguros-MNO');

INSERT INTO Mantenimiento (Fecha , Detalles , Costo , Id_Auto)

VALUES

('2024-08-05 ', 'Cambio-de-aceite ', 80, 1),
('2024-07-20 ', 'Revision-de-frenos ', 120, 2),
('2024-06-15 ', 'Alineacion-y-balanceo ', 100, 3),
('2024-05-30 ', 'Cambio-de-filtro-de-aire ', 70, 4),

```

('2024-04-10', 'Reemplazo-de-bujias', 90, 5),
('2024-03-25', 'Rotacion-de-llantas', 60, 6);
INSERT INTO Cliente (Nombre, Direccion, Telefono, Id_Auto)
VALUES
('Juan-Perez', 'Calle-123,-Ciudad-A', 555-1234, 1),
('Maria-Lopez', 'Av.-Principal,-Ciudad-B', 555-5678, 2),
('Pedro-Ramirez', 'Calle-Central,-Ciudad-C', 555-9876, 3),
('Ana-Garcia', 'Carrera-456,-Ciudad-D', 555-5432, 4),
('Luis-Torres', 'Av.-Norte,-Ciudad-E', 555-7890, 5),
('Laura-Martinez', 'Calle-Sur,-Ciudad-F', 555-2345, 6);
INSERT INTO Servicio (Fecha, Tipo, Costo, Id_Cliente)
VALUES
('2024-08-10', 'Cambio-de-aceite', 50, 1),
('2024-07-25', 'Revision-general', 80, 2),
('2024-06-18', 'Alineacion', 70, 3),
('2024-05-28', 'Cambio-de-filtro', 60, 4),
('2024-04-15', 'Reparacion-de-frenos', 100, 5),
('2024-03-30', 'Rotacion-de-llantas', 40, 6);
INSERT INTO Operador (Nombre, Licencia, Telefono, Id_Auto, Id_Servicio)
VALUES
('Carlos-Rodriguez', '123456', 555-1111, 1, 1),
('Sofia-Navarro', '654321', 555-2222, 2, 2),
('Andres-Soto', '987654', 555-3333, 3, 3),
('Elena-Vargas', '456789', 555-4444, 4, 4),
('Mario-Jimenez', '789012', 555-

```

5. Conclusiones

Identificar las principales entidades (como clientes, productos, pedidos) y sus atributos (como nombre, dirección, precio), es fundamental para comprender cómo interactúan dentro del sistema. Es crucial utilizar diagramas de entidad-relación (DER) para visualizar y organizar las entidades y sus vínculos.

Representar con claridad las interacciones entre las entidades resulta esencial.

Convertir el modelo DER en un modelo relacional puede ser complejo. Se deben crear tablas que reflejen las entidades y definir claves primarias y foráneas para mantener la integridad referencial.

Redactar sentencias SQL permite generar la base de datos, definir tablas, insertar datos y consultar su estructura. Esto incluye el uso de sistemas de gestión de bases de datos como MySQL, SQL Server o PostgreSQL.

Entender cómo los datos se relacionan dentro del sistema es indispensable. Esto es fundamental para consultas y manipulación eficaz de la información almacenada.

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.