

REPORTE DE PRÁCTICA 1.4

NOMBRE DE LA PRÁCTICA: Consultas a BD Flotillass

ALUMNO: Axel Damian Ortiz Simon
Dr. Eduardo Cornejo-Velázquez



1. Introducción

En el presente reporte de prácticas se aborda el diseño e implementación de una base de datos relacional para la gestión de flotilla de autos, utilizando la metodología sistemática y estructurada que caracteriza el modelado de bases de datos. Este proceso es fundamental, ya que permite garantizar la organización, integridad y eficiencia en el manejo de la información, aspectos esenciales en cualquier sistema de gestión. La práctica tiene como objetivo aplicar un enfoque metodológico para el diseño de la base de datos en MySQL, siguiendo las etapas clave del modelado relacional: análisis de requerimientos, diseño conceptual, diseño lógico y su posterior implementación. Este enfoque no solo asegura la correcta estructuración de los datos, sino que también facilita la solución de problemas específicos relacionados con la administración de flotillas, como el seguimiento de vehículos, mantenimiento, asignación de recursos y control de inventarios. En este contexto, se destaca la importancia de un proceso sistemático y metodológico, ya que permite desarrollar soluciones robustas y escalables que responden a las necesidades reales de la organización. La práctica no solo busca fortalecer las competencias técnicas en el uso de herramientas como MySQL, sino también fomentar habilidades analíticas y de resolución de problemas, esenciales en el ámbito profesional.

2. Marco teórico

El diseño e implementación de bases de datos relacionales es un proceso esencial en la construcción de sistemas de información, ya que permite organizar, almacenar y recuperar datos de manera eficiente. En este marco teórico se abordan los conceptos fundamentales, herramientas y metodologías relevantes para el desarrollo del proyecto de gestión de flotilla de autos, así como su importancia en el ámbito de la informática.

El álgebra relacional es un lenguaje formal que permite manipular y consultar bases de datos relacionales de manera eficiente. Es la base teórica sobre la cual se construyen lenguajes de consulta como SQL, proporcionando un marco formal para entender cómo funcionan las consultas y cómo se procesan los datos.

Su principal aplicación es servir como fundamento teórico para los lenguajes de consulta en bases de datos relacionales. Esto incluye la especificación de consultas mediante operaciones que se inspiran en la teoría de conjuntos, como unión, intersección y diferencia, así como operaciones específicas del modelo relacional, como selección, proyección y unión natural.

El álgebra relacional se inspira en la teoría de conjuntos, ya que las relaciones (tablas) se pueden considerar como conjuntos de tuplas (filas). Las operaciones del álgebra relacional, como la unión, intersección y diferencia, tienen equivalentes directos en la teoría de conjuntos, lo que refuerza su base matemática y formal.

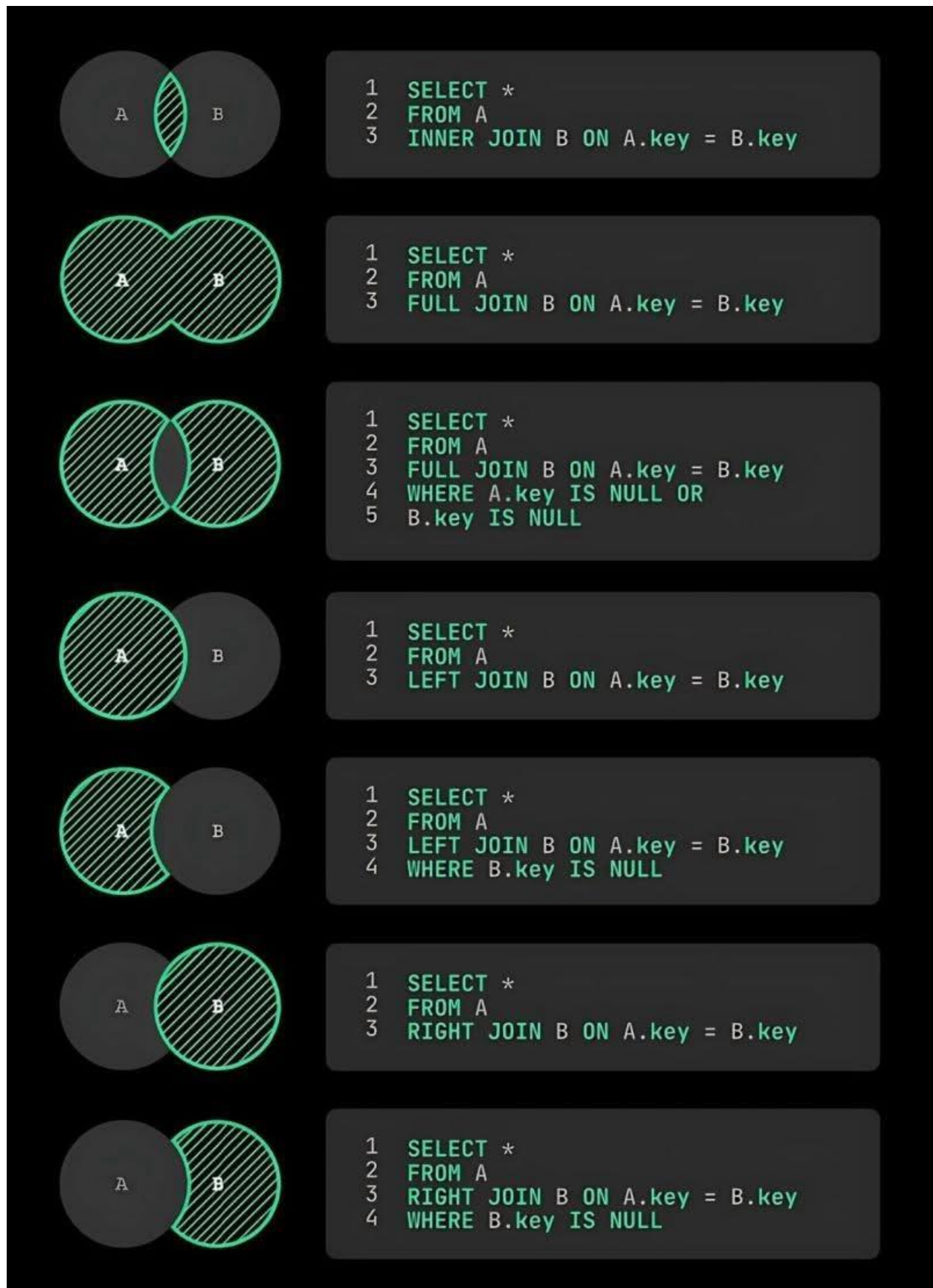


Figure 1: Algebra Re.

1. Bases de Datos Relacionales

Una base de datos relacional es una colección organizada de datos que utiliza un modelo basado en tablas para representar las relaciones entre diferentes tipos de información. Este modelo fue introducido por Edgar F. Codd en 1970 y se ha convertido en el estándar más utilizado para la gestión de datos. En el caso de la práctica, la base de datos relacional permite estructurar la información relacionada con los vehículos, conductores, mantenimientos, asignaciones y otros aspectos clave de la gestión de flotillas. Las bases de datos relacionales ofrecen ventajas significativas, como: Integridad de datos: Garantiza que los datos almacenados sean precisos y consistentes. Escalabilidad: Permite manejar grandes volúmenes de información sin comprometer el rendimiento. Flexibilidad: Facilita la modificación y actualización de los datos en función de los cambios en los requisitos del sistema.

2. Metodología para el Diseño de Bases de Datos

El diseño de una base de datos relacional implica un proceso sistemático que se compone de varias etapas: Análisis de Requerimientos: En esta fase se identifican las necesidades del sistema y los procesos que la base de datos debe soportar. Para la gestión de flotillas, esto incluye aspectos como el seguimiento de vehículos, historial de mantenimiento, asignación de conductores y costos operativos. Diseño Conceptual: Se realiza un modelo abstracto de los datos utilizando diagramas entidad-relación (ER), los cuales representan entidades (como vehículos, conductores o mantenimientos) y sus relaciones. Diseño Lógico: En esta etapa, el modelo conceptual se traduce en un esquema relacional que incluye tablas, atributos, claves primarias y claves foráneas. Implementación Física: Se lleva a cabo la creación de la base de datos en un sistema de gestión de bases de datos (SGBD), en este caso, MySQL, utilizando comandos SQL para definir y manipular tablas.

3. MySQL como Herramienta de Gestión de Bases de Datos

MySQL es un sistema de gestión de bases de datos relacional (SGBD) ampliamente utilizado en el desarrollo de aplicaciones debido a su robustez, flexibilidad y facilidad de uso. Al ser una herramienta de código abierto, MySQL se adapta bien a proyectos académicos y empresariales, ofreciendo características como: Soporte para grandes volúmenes de datos. Alta compatibilidad con otros lenguajes y plataformas. Seguridad y control de acceso

para proteger los datos sensibles.

4. Importancia del Diseño Metodológico en la Gestión de Flotillas

La gestión de flotillas de autos requiere un sistema que permita administrar eficientemente los recursos vehiculares de una organización. Entre los aspectos más relevantes destacan: Seguimiento de vehículos: Registro de ubicación, kilometraje, consumo de combustible y estado de los automóviles. Mantenimiento preventivo y correctivo: Planificación y control de las actividades de mantenimiento para garantizar la operatividad de la flotilla. Gestión de conductores: Asignación de tareas y monitoreo del desempeño. Optimización de costos: Identificación de áreas de mejora para reducir gastos operativos. Un diseño metodológico para la base de datos asegura que el sistema sea escalable, confiable y capaz de adaptarse a las necesidades cambiantes de la organización. Asimismo, permite la integración de herramientas analíticas para generar reportes e indicadores que apoyen la toma de decisiones.

Análisis de requerimientos

Requerimientos Funcionales

Gestión de Vehículos

Registro de vehículos con detalles (tipo, modelo, año, estado, consumo de combustible, capacidad de carga).

Control de mantenimiento preventivo y correctivo.

Monitoreo de documentación obligatoria (seguros, tenencia, tarjeta de circulación, verificaciones) dentro del registro del vehículo.

Gestión de Conductores Registro de información personal y profesional de los conductores. Seguimiento de licencias y permisos con alertas de vencimiento. Historial de asignación de vehículos y rutas. Gestión de Rutas y Desempeño Registro de rutas con inicio, destino, tiempos estimados y reales. Seguimiento de rutas mediante GPS y generación de alertas en caso de desvíos. Reportes de eficiencia del vehículo y desempeño del conductor. Gestión de Gastos y Consumo de Combustible Registro de consumo de combustible por vehículo y ruta. Generación de reportes detallados con costos operativos. Control del presupuesto de cada unidad. Mantenimiento y Seguridad Registro de cronogramas de mantenimiento preventivo. Historial de reparaciones y cos-

tos. Alertas automáticas para revisiones próximas. Generación de Reportes
 Reportes sobre desempeño de conductores y vehículos. Reportes de costos y
 eficiencia operativa. Historial de viajes y mantenimientos. Requerimientos
 No Funcionales Accesibilidad: Sistema disponible en computadoras y dispos-
 itivos móviles. Seguridad: Autenticación de usuarios y encriptación de datos
 sensibles. Escalabilidad: Capacidad para múltiples flotillas y usuarios. Inter-
 faz intuitiva: Adaptable para usuarios sin conocimientos técnicos. [?] para
 incorporarla al documento.

Modelo Entidad - Relación

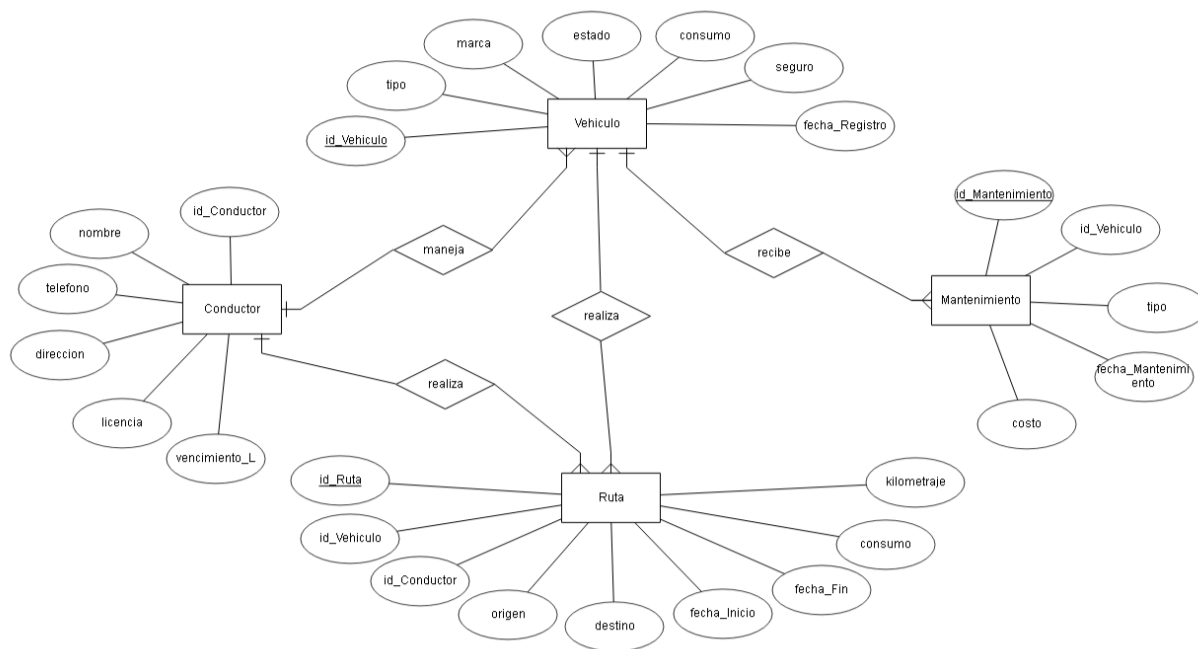


Figure 2: Modelo Entidad - Relación propuesto.

Modelo relacional

Modelo relacional

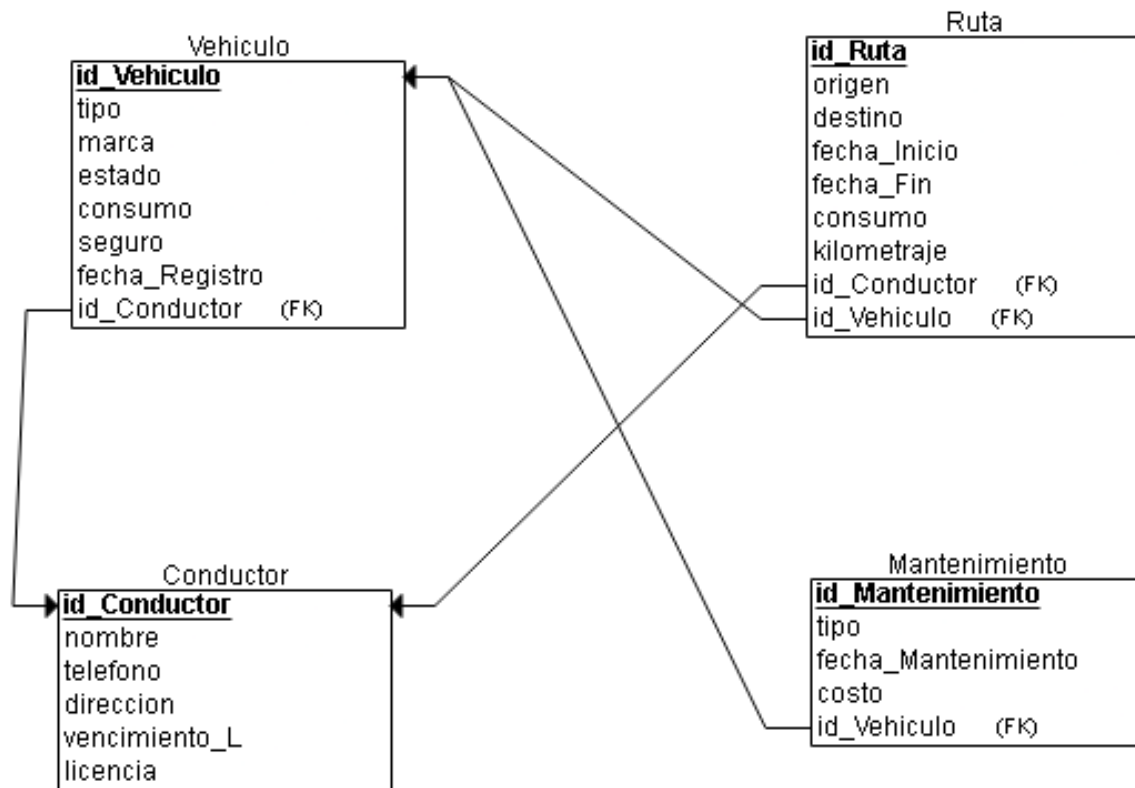


Figure 3: Esquema Relacional propuesto.

SQL ACTUALIZADA

```
2. CREATE TABLE Conductor
(
    id_Conductor INT NOT NULL,
    nombre VARCHAR(80) NOT NULL,
    telefono VARCHAR(20) NULL,
    direccion VARCHAR(80) NOT NULL,
    vencimiento_L DATE NOT NULL,
    licencia VARCHAR(30) NOT NULL,
    PRIMARY KEY (id_Conductor)
);
```

```
CREATE TABLE Vehiculo
```



```
(
    id_Vehiculo INT NOT NULL,
    tipo VARCHAR(80) NOT NULL,
    marca VARCHAR(30) NOT NULL,
    estado VARCHAR(30) NOT NULL,
    consumo FLOAT NOT NULL,
    seguro VARCHAR(50) NOT NULL,
    fecha_Registro DATE NOT NULL,
    id_Conductor INT NOT NULL,
    PRIMARY KEY (id_Vehiculo),
    FOREIGN KEY (id_Conductor)
    REFERENCES Conductor(id_Conductor)
);
```

CREATE TABLE Mantenimiento

```
(
    id_Mantenimiento INT NOT NULL,
    tipo VARCHAR(50) NOT NULL,
    fecha_Mantenimiento DATE NOT NULL,
    costo FLOAT NOT NULL,
    id_Vehiculo INT NOT NULL,
    PRIMARY KEY (id_Mantenimiento),
    FOREIGN KEY (id_Vehiculo)
    REFERENCES Vehiculo(id_Vehiculo)
);
```

CREATE TABLE Ruta

```
(
    id_Ruta INT NOT NULL,
    origen VARCHAR(50) NOT NULL,
    destino VARCHAR(50) NOT NULL,
    fecha_Inicio DATE NOT NULL,
    fecha_Fin DATE NOT NULL,
    consumo FLOAT NOT NULL,
    kilometraje FLOAT NOT NULL,
    id_Conductor INT NOT NULL,
```

```
id_Vehiculo INT NOT NULL,  
PRIMARY KEY (id_Ruta),  
FOREIGN KEY (id_Conductor)  
REFERENCES Conductor(id_Conductor),  
FOREIGN KEY (id_Vehiculo)  
REFERENCES Vehiculo(id_Vehiculo)  
);
```

```
INSERT INTO Conductor (id_Conductor, nombre, telefono,  
direccion, vencimiento_L, licencia) VALUES  
(1, 'Carlos Ruiz', 5523412345, 'Av. Principal 123, Pachuca',  
'2026-05-15', 'A1234567890'),  
(2, 'María López', 5523456789, 'Calle Secundaria 456, Pachuca',  
'2025-11-20', 'B9876543210'),  
(3, 'Juan Pérez', 5523987654, 'Boulevard Centro 789, Pachuca',  
'2027-02-10', 'C5678901234'),  
(4, 'Ana Martínez', 5523123456, 'Avenida Universidad 321,  
Pachuca', '2024-12-05', 'D3456789012'),  
(5, 'Luis Sánchez', 5523345678, 'Calle Comercio 654, Pachuca',  
'2025-08-25', 'E4567890123');
```

```
INSERT INTO Vehiculo  
(id_Vehiculo, tipo, marca,  
estado, consumo,  
seguro, fecha_Registro, id_Conductor) V  
ALUES  
(1, 'Sedan', 'Toyota', 'Activo', 15.5, 'Seguros S.A.',  
'2023-01-10', 1),  
(2, 'Camioneta', 'Ford', 'En reparación', 12.0, 'Protección S.A.',  
'2023-03-15', 2),  
(3, 'SUV', 'Honda', 'Activo', 14.8, 'Seguridad S.A.',  
'2023-05-20', 3),  
(4, 'Coupé', 'Chevrolet', 'En mantenimiento', 13.2,  
'Aseguradora S.A.',  
'2023-07-25', 4),  
(5, 'Hatchback', 'Volkswagen', 'Activo', 16.0, 'Progreso S.A.',
```

```
'2023-09-30', 5);
```

```
INSERT INTO Vehiculo (id_Vehiculo, tipo, marca, estado,
consumo, seguro, fecha_Registro, id_Conductor) VALUES
(6, 'Sedan', 'Mazda', 'Activo', 18.2, 'Seguros Globales',
'2025-01-05', 6),
(7, 'Camioneta', 'Chevrolet', 'En reparación', 17.4,
'Aseguradora Nacional',
'2025-01-10', 7),
(8, 'SUV', 'BMW', 'Activo', 19.0, 'Seguros A1',
'2025-01-15', 8),
(9, 'Coupé', 'Mercedes', 'En mantenimiento', 20.5,
'Mundo Seguros',
'2025-01-20', 9),
(10, 'Hatchback', 'Ford', 'Activo', 21.0, 'Seguros Rápidos',
'2025-01-25', 10),
(11, 'Sedan', 'Nissan', 'Activo', 18.5, 'Seguro Total',
'2025-01-30', 11),
(12, 'Camioneta', 'Toyota', 'En reparación', 19.8,
'Seguros del Norte',
'2025-02-02', 12),
(13, 'SUV', 'Audi', 'Activo', 22.0, 'AseguraTodo',
'2025-02-05', 13),
(14, 'Coupé', 'BMW', 'En mantenimiento', 20.2,
'Protección Integral',
'2025-02-10', 14),
(15, 'Hatchback', 'Volkswagen', 'Activo', 23.0,
'Aseguradora Vida',
'2025-02-15', 15);
```

```
INSERT INTO Mantenimiento (id_Mantenimiento, tipo,
fecha_Mantenimiento, costo, id_Vehiculo) VALUES
(1, 'Cambio de aceite', '2024-02-15', 1200.50, 1),
(2, 'Revisión general', '2024-04-20', 3000.00, 2),
(3, 'Cambio de llantas', '2024-06-25', 4500.75, 3),
```

```
(4, 'Reemplazo de frenos', '2024-08-30', 3500.20, 4),  
(5, 'Alineación y balanceo', '2024-10-05', 1800.90, 5);
```

```
INSERT INTO Ruta  
(id_Ruta, origen, destino, fecha_Inicio,  
fecha_Fin, consumo, kilometraje, id_Conductor, id_Vehiculo)  
VALUES  
(1, 'Pachuca', 'Ciudad de México', '2024-03-01', '2024-03-01',  
15.5, 94.0, 1, 1),  
(2, 'Pachuca', 'Querétaro', '2024-05-10',  
'2024-05-10', 14.0, 235.0, 2, 2),  
(3, 'Pachuca', 'Toluca', '2024-07-15',  
'2024-07-15', 13.5, 120.0, 3, 3),  
(4, 'Pachuca', 'Puebla', '2024-09-20',  
'2024-09-20', 12.0, 160.0, 4, 4),  
(5, 'Pachuca', 'Guadalajara', '2024-11-25',  
'2024-11-25', 16.5, 500.0, 5, 5);
```

```
INSERT INTO Ruta  
(id_Ruta, origen, destino, fecha_Inicio, fecha_Fin,  
consumo, kilometraje, id_Conductor, id_Vehiculo)  
VALUES  
(6, 'Pachuca', 'Monterrey', '2025-02-19',  
'2025-02-19', 20.5, 800.0, 1, 2),  
(7, 'Ciudad de México', 'León', '2025-02-18',  
'2025-02-19', 18.0, 380.0, 2, 3),  
(8, 'Querétaro', 'Guadalajara', '2025-02-19',  
'2025-02-20', 16.5, 400.0, 3, 4),  
(9, 'Toluca', 'Morelia', '2025-02-19',  
'2025-02-19', 14.0, 250.0, 4, 1),  
(10, 'Puebla', 'Veracruz', '2025-02-18',  
'2025-02-19', 17.5, 320.0, 5, 5);
```

```
INSERT INTO Ruta  
(id_Ruta, origen, destino, fecha_Inicio  
, fecha_Fin, consumo, kilometraje, id_Conductor, id_Vehiculo)
```

VALUES

```
(11, 'Aguascalientes', 'Zacatecas', '2025-02-15',  
'2025-02-15', 18.0, 350.0, 6, 6),  
(12, 'Ciudad de México', 'Puebla', '2025-02-16',  
'2025-02-16', 19.5, 360.0, 7, 7),  
(13, 'Guadalajara', 'León', '2025-02-17',  
'2025-02-17', 17.8, 400.0, 8, 8),  
(14, 'Monterrey', 'Saltillo', '2025-02-18',  
'2025-02-18', 20.0, 320.0, 9, 9),  
(15, 'Toluca', 'Pachuca', '2025-02-19',  
'2025-02-19', 16.5, 280.0, 10, 10),  
(16, 'Querétaro', 'San Luis Potosí',  
'2025-02-20', '2025-02-20', 18.8, 410.0, 11, 11),  
(17, 'Mérida', 'Cancún', '2025-02-21',  
'2025-02-21', 19.2, 450.0, 12, 12),  
(18, 'Oaxaca', 'Veracruz', '2025-02-22',  
'2025-02-22', 21.5, 390.0, 13, 13),  
(19, 'Chetumal', 'Campeche', '2025-02-23',  
'2025-02-23', 22.0, 500.0, 14, 14),  
(20, 'Hermosillo', 'Mexicali', '2025-02-24',  
'2025-02-24', 23.0, 600.0, 15, 15);
```

```
INSERT INTO Mantenimiento (id_Mantenimiento,  
tipo, fecha_Mantenimiento, costo, id_Vehiculo) VALUES  
(6, 'Cambio de batería', '2023-11-10', 2500.00, 1),  
(7, 'Revisión de motor', '2023-12-05', 3200.75, 2),  
(8, 'Cambio de aceite', '2024-01-18', 1300.00, 3),  
(9, 'Cambio de frenos', '2024-03-25', 4000.50, 4),  
(10, 'Alineación y balanceo', '2024-06-30', 1700.25, 5),  
(11, 'Cambio de llantas', '2024-09-12', 5000.00, 1),  
(12, 'Revisión general', '2024-12-28', 2800.90, 2),  
(13, 'Cambio de aceite', '2025-01-05', 1400.75, 3),  
(14, 'Revisión de suspensión', '2025-01-15', 3500.00, 4),  
(15, 'Cambio de luces', '2025-02-10', 900.50, 5);
```

```
INSERT INTO Verificacion (fecha_Verificacion, resultado, id_Vehiculo)
VALUES
```

```
('2025-03-05', 'Aprobado', 1),
('2025-03-10', 'Pendiente', 2),
('2025-03-15', 'Aprobado', 3);
```

```
CREATE TABLE Refaccion (
    id_Refaccion INT NOT NULL AUTO_INCREMENT,
    nombre VARCHAR(80) NOT NULL,
    costo FLOAT NOT NULL,
    cantidad INT NOT NULL,
    id_Mantenimiento INT NOT NULL,
    PRIMARY KEY (id_Refaccion),
    FOREIGN KEY (id_Mantenimiento)
    REFERENCES Mantenimiento(id_Mantenimiento)
);
```

```
ALTER TABLE Refaccion ADD COLUMN fecha_uso DATE;
```

```
INSERT INTO Refaccion (nombre, costo,
cantidad, id_Mantenimiento) VALUES
('Filtro de aceite', 350.00, 1, 1),
('Bujías', 600.00, 4, 2),
('Pastillas de freno', 1200.00, 2, 3),
('Batería', 2500.00, 1, 4),
('Amortiguadores', 1800.00, 2, 5),
('Filtro de aire', 400.00, 1, 6),
('Aceite sintético', 900.00, 5, 7),
('Correa de distribución', 3200.00, 1, 8),
('Llanta nueva', 2500.00, 1, 9),
('Radiador', 4500.00, 1, 10);
```

```
SELECT V.id_Vehiculo, V.tipo, V.marca,
M.id_Mantenimiento, M.tipo AS fecha_mantenimiento,
M.fecha_Mantenimiento, M.costo
```

```
FROM Vehiculo as V
JOIN Mantenimiento M ON V.id_Vehiculo =
M.id_Vehiculo
WHERE year(M.fecha_Mantenimiento)=2
025 and month(fecha_Mantenimiento)=1;
```

```
SELECT V.id_Vehiculo, V.tipo, V.marca,
Ver.fecha_Verificacion, Ver.resultado
FROM Vehiculo V
JOIN Verificacion Ver ON V.id_Vehiculo
= Ver.id_Vehiculo
WHERE Ver.fecha_Verificacion BETWEEN '
2025-03-01' AND '2025-03-31';
```

```
SELECT C.nombre, R.id_Ruta, R.origen, R.destino,
R.fecha_Inicio, R.fecha_Fin
FROM Conductor C
JOIN Ruta R ON C.id_Conductor = R.id_Conductor
WHERE '2025-02-19' BETWEEN R.fecha_Inicio AND R.fecha_Fin;
```

```
SELECT
    V.id_Vehiculo,
    V.tipo,
    V.marca,
    R.consumo,
    (R.consumo * 20) AS costo_gasolina_diario
FROM Vehiculo V JOIN Ruta R ON V.id_Vehiculo =
R.id_Vehiculo
WHERE R.fecha_Inicio >= CURDATE() -
INTERVAL 2 MONTH AND (R.consumo * 20) > 2300;
```

```
SELECT nombre AS Refaccion, SUM(costo * cantidad) AS Total_Gasto
FROM Refaccion
WHERE fecha_uso BETWEEN '2025-01-01' AND '2025-01-31'
GROUP BY nombre;
```

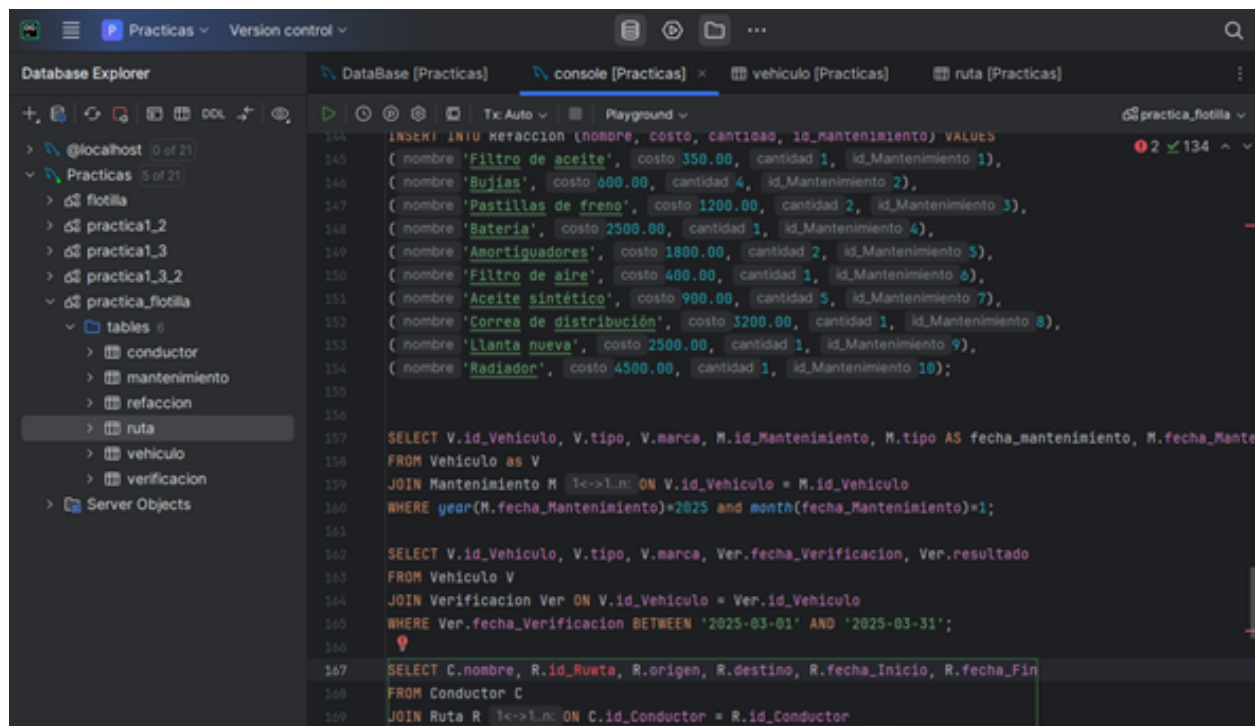


Figure 4: Base de Datos.

3. Herramientas empleadas

Describir qué herramientas se han utilizado...

1. ERD Plus: ERD Plus es una herramienta en línea para el diseño de diagramas de entidad-relación (ERD). Se utiliza para modelar la estructura de una base de datos, mostrando las entidades, sus atributos y las relaciones entre ellas. Es muy útil para visualizar y planificar la organización de datos antes de implementarlos en un sistema de gestión de bases de datos (DBMS).
2. DataGrip: DataGrip es un entorno de desarrollo integrado (IDE) para bases de datos desarrollado por JetBrains. Soporta múltiples sistemas de gestión de bases de datos, incluyendo MySQL, PostgreSQL, SQL Server, entre otros. Esta herramienta facilita la escritura y ejecución de consultas SQL, el diseño y administración de esquemas, la gestión de conexiones y el análisis del rendimiento de las consultas. Su interfaz intuitiva y sus potentes funcionalidades hacen que sea una opción ideal para desarrolladores y administradores de bases de datos.
3. LaTeX: LaTeX es un sistema de preparación de documentos utilizado

principalmente para la creación de documentos científicos y técnicos. Permite escribir texto, fórmulas matemáticas y otros elementos de manera estructurada y profesional. LaTeX es especialmente útil para la producción de documentos con un alto estándar tipográfico y es ampliamente utilizado en la academia y la investigación.

4. Consultas

1. Listado de autos que recibieron mantenimiento en enero del 2025.

$$\pi_{id_Vehiculo, tipo, marca, id_Mantenimiento, tipo, fecha_Mantenimiento, costo} \left(\sigma_{year(fecha_Mantenimiento)=2025 \wedge month(fecha_Mantenimiento)=1} (Vehiculo \times Mantenimiento) \right)$$

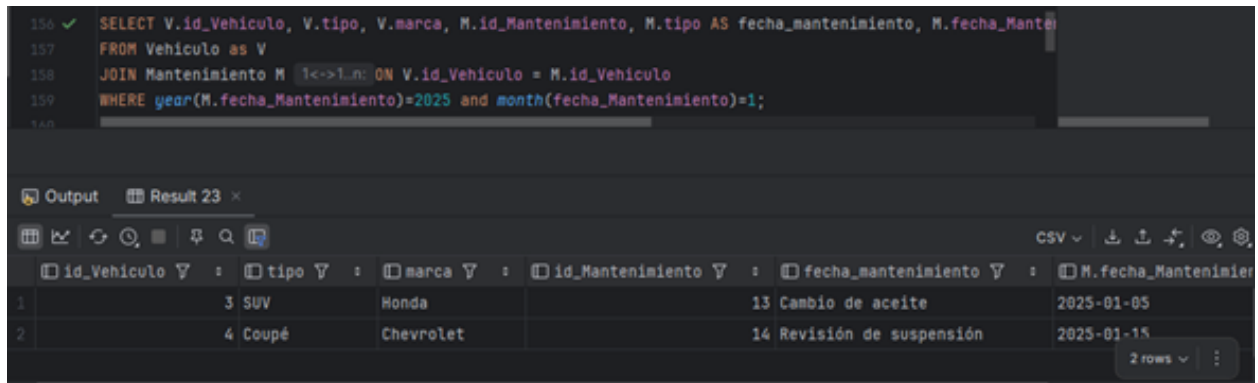


Figure 5: Consulta 1.

2. Listado de autos que deben ser verificados en marzo del 2025.

$$\pi_{id_Vehiculo, tipo, marca, fecha_Verificacion, resultado} \left(\sigma_{fecha_Verificacion \geq '2025-03-01' \wedge fecha_Verificacion \leq '2025-03-31'} (Vehiculo \times Verificacion) \right)$$

The screenshot shows a SQL query in a dark-themed editor. The query is as follows:

```

161 ✓ SELECT V.id_Vehiculo, V.tipo, V.marca, Ver.fecha_Verificacion, Ver.resultado
162 FROM Vehiculo V
163 JOIN Verificacion Ver ON V.id_Vehiculo = Ver.id_Vehiculo
164 WHERE Ver.fecha_Verificacion BETWEEN '2025-03-01' AND '2025-03-31';
165

```

Below the query, the 'Output' tab shows 'Result 25' with a table of 3 rows and 5 columns. The columns are: id_Vehiculo, tipo, marca, fecha_Verificacion, and resultado. The data is as follows:

	id_Vehiculo	tipo	marca	fecha_Verificacion	resultado
1	1	Sedan	Toyota	2025-03-05	Aprobado
2	2	Camioneta	Ford	2025-03-10	Pendiente
3	3	SUV	Honda	2025-03-15	Aprobado

Figure 6: Consulta 2.

3.Reporte con el nombre de los conductores que trabajaron el 14 de febrero del 2025.

$$\pi_{\text{nombre}, \text{id_Ruta}, \text{origen}, \text{destino}, \text{fecha_Inicio}, \text{fecha_Fin}} \left(\sigma_{\text{fecha_Inicio} \leq '2025-02-19' \wedge \text{fecha_Fin} \geq '2025-02-14'} (\text{Conductor} \times \text{Ruta}) \right)$$

The screenshot shows a SQL query in a dark-themed editor. The query is as follows:

```

166 ✓ SELECT C.nombre, R.id_Ruta, R.origen, R.destino, R.fecha_Inicio, R.fecha_Fin
167 FROM Conductor C
168 JOIN Ruta R ON C.id_Conductor = R.id_Conductor
169 WHERE '2025-02-14' BETWEEN R.fecha_Inicio AND R.fecha_Fin;
170

```

Below the query, the 'Output' tab shows 'Result 28' with a table of 3 rows and 6 columns. The columns are: nombre, id_Ruta, origen, destino, fecha_Inicio, and fecha_Fin. The data is as follows:

	nombre	id_Ruta	origen	destino	fecha_Inicio	fecha_Fin
1	Carlos Ruiz	6	Pachuca	Monterrey	2025-02-14	2025-02-14
2	María López	7	Ciudad de México	León	2025-02-14	2025-02-14
3	Juan Pérez	8	Querétaro	Guadalajara	2025-02-14	2025-02-14

Figure 7: Consulta 3.

4.Listado de autos que consumieron más de 2,300.00 de gasolina por día en

los últimos 2 meses.

$$\pi_{id_Vehiculo, tipo, marca, consumo, \text{costo_gasolina_diario}} \left(\sigma_{\text{fecha_Inicio} \geq \text{CURDATE()} - \text{INTERVAL } 2 \text{ MONTH} \wedge (\text{consumo} \times 20) > 2300} (\text{Vehiculo} \times \text{Ruta}) \right)$$

```

171 ✓ SELECT
172     V.id_Vehiculo,
173     V.tipo,
174     V.marca,
175     R.consumo,
176     (R.consumo * 20) AS costo_gasolina_diario
177 FROM Vehiculo V JOIN Ruta R ON V.id_Vehiculo = R.id_Vehiculo
178 WHERE R.fecha_Inicio >= CURDATE() - INTERVAL 2 MONTH AND (R.consumo * 20) > 2300;
179

```

	id_Vehiculo	tipo	marca	consumo	costo_gasolina_diario
1	2	Camioneta	Ford	120	2400
2	3	SUV	Honda	120	2400
3	4	Coupé	Chevrolet	120	2400
4	1	Sedan	Toyota	120	2400

Figure 8: Consulta 4.

5. Reporte de gastos en refacciones utilizadas en el mes de enero del 2025.

$$\pi_{\text{nombre}, \text{Total_Gasto}} \left(\gamma_{\text{nombre}; \sum(\text{costo} \times \text{cantidad}) \rightarrow \text{Total_Gasto}} \left(\sigma_{\text{fecha_uso} \geq '2025-01-01' \wedge \text{fecha_uso} \leq '2025-01-31'} (\text{Refaccion}) \right) \right)$$

```
180 ✓ SELECT nombre AS Refaccion, SUM(costo * cantidad) AS Total_Gasto
181 FROM Refaccion
182 WHERE fecha_uso BETWEEN '2025-01-01' AND '2025-01-31'
183 GROUP BY nombre;
184
185
```

Output Result 30

	Refaccion ▼	Total_Gasto ▼
1	Filtro de aceite	700
2	Batería	5000
3	Filtro de aire	800
4	Radiador	9000

Console [Practicas]

Figure 9: Consulta 5.

6. Obtener los nombres de los conductores y los vehículos que manejan

$$\pi_{\text{nombre}, \text{id_Vehiculo}, \text{tipo}, \text{marca}} \left((\text{Conductor} \bowtie \text{Vehiculo}) \right)$$

```

187 ✓ SELECT C.nombre, V.id_Vehiculo, V.tipo, V.marca
188 FROM Conductor C
189 JOIN Vehiculo V ON C.id_Conductor = V.id_Vehiculo;
190

```

Output Result 39 ×

	nombre ▾	:	id_Vehiculo ▾	:	tipo ▾	:	marca ▾	:
1	Carlos Ruiz				1 Sedan		Toyota	
2	María López				2 Camioneta		Ford	
3	Juan Pérez				3 SUV		Honda	
4	Ana Martínez				4 Coupé		Chevrolet	

Console [Practicas] 178:1 CRLF U

Figure 10: Consulta 6.

7. Vehículos con un mantenimiento específico de "Cambio de aceite"

$$\pi_{id_Vehiculo, tipo, marca, fecha_Mantenimiento, costo} \left(\sigma_{tipo='Cambio de aceite'} (Vehiculo \times Mantenimiento) \right)$$

```

191 ✓ SELECT V.id_Vehiculo, V.tipo, V.marca, M.fecha_Mantenimiento, M.costo
192 FROM Vehiculo V
193 JOIN Mantenimiento M 1<->1 ON V.id_Vehiculo = M.id_Vehiculo
194 WHERE M.tipo = 'Cambio de aceite';
195

```

Output Result 40 ×

	id_Vehiculo ▾	:	tipo ▾	:	marca ▾	:	fecha_Mantenimiento ▾	:	costo ▾	:
1			1 Sedan		Toyota		2024-02-15		1200.5	
2			3 SUV		Honda		2024-01-18		1300	
3			3 SUV		Honda		2025-01-05		1400.75	

CSV ▾ ⬇

Figure 11: Consulta 7.

8. Mostrar el costo total de mantenimiento de cada vehículo

$$\gamma_{id_Vehiculo, tipo, marca; \sum(costo) \rightarrow Total_Mantenimiento} (Vehiculo \bowtie Mantenimiento)$$

```

196 ✓ SELECT V.id_Vehiculo, V.tipo, V.marca, SUM(M.costo) AS Total_Mantenimiento
197 FROM Vehiculo V
198 JOIN Mantenimiento M 1<->1..n: ON V.id_Vehiculo = M.id_Vehiculo
199 GROUP BY V.id_Vehiculo, V.tipo, V.marca;

```

	id_Vehiculo	tipo	marca	Total_Mantenimiento
1	1	Sedan	Toyota	8700.5
2	2	Camioneta	Ford	9001.64990234375
3	3	SUV	Honda	7201.5
4	4	Coupé	Chevrolet	11000.699951171875

Figure 12: Consulta 8.

9. Obtener la cantidad de rutas realizadas por cada conductor

$$\gamma_{\text{nombre}; \text{COUNT}(\text{id_Ruta}) \rightarrow \text{Total_Rutas}}(\text{Conductor} \bowtie \text{Ruta})$$

```

201 ✓ SELECT C.nombre, COUNT(R.id_Ruta) AS Total_Rutas
202 FROM Conductor C
203 JOIN Ruta R 1<->1..n: ON C.id_Conductor = R.id_Conductor
204 GROUP BY C.nombre;
205

```

	nombre	Total_Rutas
1	Carlos Ruiz	3
2	María López	4
3	Juan Pérez	4
4	Ana Martínez	4

Figure 13: Consulta 9.

10. Obtener los vehículos que han recorrido más de 1000 km en total

$$\sigma_{\text{Total_Kilometros} > 1000} \left(\gamma_{\text{id_Vehiculo, tipo, marca; } \sum(\text{kilometraje}) \rightarrow \text{Total_Kilometros}}(\text{Vehiculo} \bowtie \text{Ruta}) \right)$$

```
206 ✓ SELECT V.id_Vehiculo, V.tipo, V.marca, SUM(R.kilometraje) AS Total_Kilometros
207 FROM Vehiculo V
208 JOIN Ruta R 1<->1.n: ON V.id_Vehiculo = R.id_Vehiculo
209 GROUP BY V.id_Vehiculo, V.tipo, V.marca
210 HAVING SUM(R.kilometraje) > 1000;
211
```

Output Result 43 x

	id_Vehiculo ▾	:	tipo ▾	:	marca ▾	:	Total_Kilometros ▾	:
1			2 Camioneta		Ford		1035	

Figure 14: Consulta 10.

5. Conclusiones

La práctica constante con consultas en álgebra relacional me ha permitido mejorar mi comprensión sobre la manipulación de bases de datos. A medida que he trabajado en la formulación de diferentes consultas, he podido identificar patrones en las operaciones básicas como selección, proyección, unión y agregación, lo que ha facilitado mi capacidad para estructurar consultas más complejas de manera lógica y eficiente.

Además, al representar estas consultas en notación matemática y en LaTeX, he reforzado mi habilidad para traducir problemas reales en expresiones formales, lo que no solo me ha ayudado a entender mejor los fundamentos del álgebra relacional, sino que también me ha preparado para trabajar con SQL y otras herramientas de gestión de bases de datos en el futuro.

Aún hay margen de mejora, especialmente en la optimización de consultas y en la combinación de múltiples relaciones para obtener información específica, pero cada ejercicio realizado ha contribuido significativamente a mi aprendizaje. Esta experiencia ha sido fundamental para desarrollar una mayor intuición sobre cómo estructurar y analizar consultas en bases de datos relacionales.

Referencias Bibliográficas

References

- [1] oronel, C., Morris, S., Rob, P. (**2020**). *Database Systems: Design, Implementation, and Management* (13^a ed.). Cengage Learning.
- [2] lmasri, R., Navathe, S. (**2016**). *Fundamentals of Database Systems* (7^a ed.). Pearson.
- [3] uBois, P. (**2008**). *MySQL: The Complete Reference* (2^a ed.). McGraw-Hill Education.
- [4] hiani, G., Laporte, G., Musmanno, R. (**2013**). *Introduction to Logistics Systems Management* (2^a ed.). Wiley.
- [5] ernandez, M. J. (**2013**). *Database Design for Mere Mortals* (3^a ed.). Addison-Wesley.