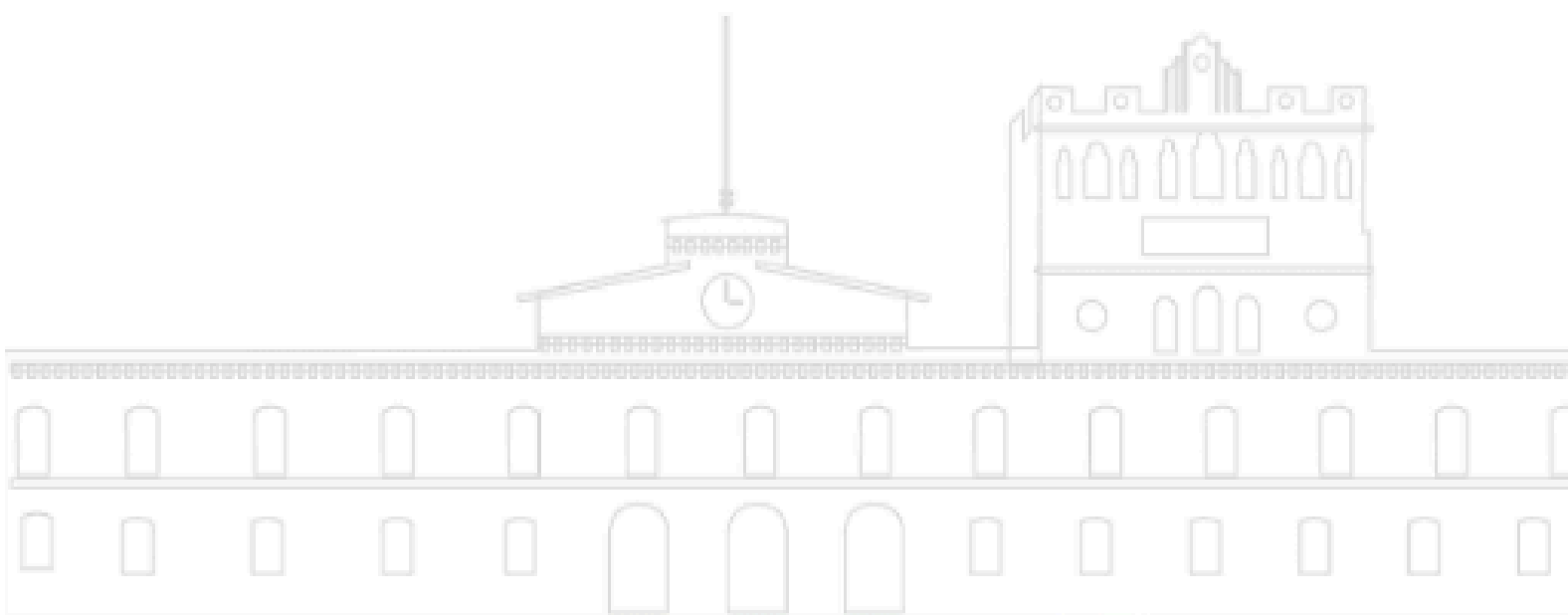


Reporte de Practica No. 2.1

ALUMNO: Cesar Rodriguez Garcia
Dr. Eduardo Cornejo-Velázquez



1. Introducción

La gestión eficaz de un parque automotor es crucial para el rendimiento logístico y la solidez financiera de cualquier empresa de transporte o distribución. Los sistemas de gestión dispersos o ineficientes suelen generar una escalada en los costos operacionales, incumplimientos regulatorios y una caída en la productividad. La solución a estos retos radica en la **consolidación y el análisis estructurado de datos** a través de una plataforma de base de datos bien diseñada.

Este documento presenta el desarrollo de una base de datos relacional implementada en **MySQL**, diseñada para gestionar integralmente una flota de vehículos. El propósito fundamental es establecer una herramienta central que no solo garantice la **integridad, seguridad y disponibilidad de los datos**, sino que también facilite el acceso controlado y específico a diversos actores clave: personal administrativo, centros de servicio técnico, operadores de vehículos y dueños de los activos.

El proceso se adhirió a una metodología rigurosa: desde la recopilación de requerimientos operacionales, la elaboración de un **Modelo Entidad-Relación (MER)**, la formalización de un esquema lógico **normalizado**, hasta su despliegue físico. Adicionalmente, se incorporó un robusto esquema de seguridad que utiliza **vistas SQL** para lograr una "segmentación virtual" de los datos, asegurando que cada rol acceda exclusivamente a la información pertinente a sus responsabilidades.

2. Fundamentos Teóricos

2.1. El Modelo Relacional y su Aplicación

El modelo de datos relacional, basado en la teoría de conjuntos, organiza la información en tablas conectadas mediante atributos comunes. Esta arquitectura es fundamental para el proyecto debido a su capacidad para garantizar la **consistencia** y la facilidad en la ejecución de **consultas complejas** a través del lenguaje SQL, minimizando la redundancia de datos.

2.2. Plataforma de Implementación: MySQL

Se seleccionó MySQL, un sistema de código abierto, reconocido por su **alto rendimiento, escalabilidad y compatibilidad** con el estándar SQL. Su uso permite una implementación robusta que asegura las propiedades de **atomicidad, consistencia, aislamiento y durabilidad (ACID)** para todas las transacciones de gestión vehicular.

2.3. Estructuración del Diseño de la Base de Datos

El diseño se desarrolló en tres etapas fundamentales:

1. **Conceptual:** Definición de los principales elementos (entidades), sus características (atributos) y los vínculos entre ellos, resultando en un MER independiente de la tecnología.

2. **Lógica:** Conversión del MER en un esquema de tablas, definiendo **claves primarias (PK)** y **claves foráneas (FK)**, y aplicando el proceso de normalización.
3. **Física:** Decisiones específicas del RDBMS (MySQL) para la optimización de desempeño, como la selección de tipos de datos e indexación.

2.4. Normalización de Datos

La normalización se aplicó para estructurar las tablas, reduciendo la redundancia y mejorando la integridad. Se implementó un diseño que cumple con la **Tercera Forma Normal (3NF)**, asegurando que la dependencia de cada campo sea exclusiva de la clave primaria de su tabla.

2.5. Seguridad y Control de Acceso Mediante Vistas

Para implementar el control de acceso, se empleó la técnica de **fragmentación virtual** a través de **Vistas SQL**. Una vista funciona como una tabla virtual, permitiendo exponer solo un subconjunto específico de datos a un usuario o rol sin concederle permisos directos sobre las tablas base. Esta estrategia asegura el **principio del mínimo privilegio**.

3. Metodología de Desarrollo

3.1. Requisitos Operacionales

El análisis funcional se centró en las siguientes necesidades de gestión:

- **Entidades Centrales:** Vehículo, Conductor, Dueño del Activo, Taller de Mantenimiento, Registro de Ruta, Transacciones y Documentos Legales.
- **Funcionalidades Clave:** Seguimiento detallado de egresos (peajes, combustible), monitoreo de vencimientos (licencias, seguros), y registro integral de servicios (preventivos y correctivos).
- **Perfiles de Acceso (Roles):**
 - **Administrativo:** Vista global para control de costos, estado de toda la flota y alertas.
 - **Taller:** Acceso restringido al historial técnico de las unidades que atienden.
 - **Conductor:** Consulta de itinerarios asignados y registro de gastos operacionales.
 - **Dueño:** Visibilidad sobre el rendimiento, costos de servicio y estado legal de sus activos.

3.2. Estructura Lógica Normalizada

El modelo conceptual se tradujo en el siguiente esquema relacional, con énfasis en la 3NF:

- **Dueño_Activo** (id Dueño, nombre, rfc, contacto)
- **Unidad_Movil** (\$\text{idUnidad}\$, \$\text{placa}\$, marca, modelo, \$\text{año}\$, \$\text{nSerie}\$, \$\text{idDueño}\$)

- **Personal_Operativo** (\$\text{idConductor}\$, nombre, \$\text{licencia}\$, \$\text{vencimientoLicencia}\$)
- **Centro_Servicio** (\$\text{idTaller}\$, nombre, dirección, teléfono)
- **Servicio_Mantenimiento** (\$\text{idMantenimiento}\$, \$\text{idUnidad}\$, \$\text{idTaller}\$, \$\text{fechaServicio}\$, \$\text{detalle}\$, \$\text{costo}\$)
- **Documentacion** (\$\text{idDoc}\$, \$\text{idUnidad}\$, \$\text{tipoDoc}\$, \$\text{vencimiento}\$)
- **Transaccion** (\$\text{idGasto}\$, \$\text{idUnidad}\$, \$\text{idConductor}\$, \$\text{concepto}\$, \$\text{monto}\$, \$\text{fecha}\$)
- **Ruta_Asignada** (\$\text{idRuta}\$, \$\text{idUnidad}\$, \$\text{idConductor}\$, \$\text{origen}\$, \$\text{destino}\$, \$\text{fechaInicio}\$, \$\text{estado}\$)

3.3. Implementación de la Seguridad

La seguridad se implementó a través de Vistas para la segmentación de datos. Solo estas vistas reciben los permisos de consulta (GRANT SELECT) para cada rol, garantizando la privacidad y el control de la información sensible.

4. Desarrollo Físico (MySQL)

4.1. Definición del Esquema (DDL - Fragmento)

El código SQL DDL creó la estructura base y las claves foráneas que garantizan la integridad referencial.

```
CREATE DATABASE IF NOT EXISTS control_logistico;
```

```
USE control_logistico;
```

```
CREATE TABLE Dueño_Activo (
```

```
    idDueño INT PRIMARY KEY AUTO_INCREMENT,
```

```
    nombre VARCHAR(100) NOT NULL,
```

```
    rfc VARCHAR(13) UNIQUE
```

```
);
```

```
CREATE TABLE Unidad_Movil (
```

```
    idUnidad INT PRIMARY KEY AUTO_INCREMENT,
```

```
    placa VARCHAR(10) UNIQUE NOT NULL,
```

```

    marca VARCHAR(50),
    modelo VARCHAR(50),
    anio INT,
    nSerie VARCHAR(20) UNIQUE,
    idDueño INT,
    FOREIGN KEY (idDueño) REFERENCES Dueño_Activo(idDueño)
);

```

-- ... se crearon las tablas restantes con sus respectivas FKs.

4.2. Vistas de Acceso por Rol (Fragmentos)

Vista para el Rol Administrativo

Diseñada para una supervisión integral, combinando información de activos, personal y estado operacional.

```

CREATE OR REPLACE VIEW V_Supervisor_Flota AS
SELECT
    um.placa, um.marca, um.modelo,
    po.nombre AS conductor_asignado,
    ra.origen, ra.destino, ra.estado AS estado_ruta,
    sm.fechaServicio
FROM Unidad_Movil um
LEFT JOIN Ruta_Asignada ra ON um.idUnidad = ra.idUnidad
LEFT JOIN Personal_Operativo po ON ra.idConductor = po.idConductor
LEFT JOIN Servicio_Mantenimiento sm ON um.idUnidad = sm.idUnidad;

```

-- Ejemplo de uso:

```

SELECT placa, estado_ruta FROM V_Supervisor_Flota WHERE estado_ruta =
'Activa';

```

Vista para el Centro de Servicio (Taller)

Restringida a los datos técnicos y el historial de reparaciones para un taller específico (ej. ID 5).

```
CREATE OR REPLACE VIEW V_Servicio_Tecnico_5 AS

SELECT

    um.placa, um.marca, um.modelo, um.anio, um.nSerie,

    sm.fechaServicio, sm.detalle, sm.costo

FROM Servicio_Mantenimiento sm

JOIN Unidad_Movil um ON sm.idUnidad = um.idUnidad

WHERE sm.idTaller = 5;
```

-- Ejemplo de uso:

```
SELECT placa, costo FROM V_Servicio_Tecnico_5 ORDER BY costo DESC;
```

Sistema Centralizado para la Administración de Activos Vehiculares: Diseño e Implementación en MySQL

1. Introducción

La gestión eficaz de un parque automotor es crucial para el rendimiento logístico y la solidez financiera de cualquier empresa de transporte o distribución. Los sistemas de gestión dispersos o ineficientes suelen generar una escalada en los costos operacionales, incumplimientos regulatorios y una caída en la productividad. La solución a estos retos radica en la **consolidación y el análisis estructurado de datos** a través de una plataforma de base de datos bien diseñada.

Este documento presenta el desarrollo de una base de datos relacional implementada en **MySQL**, diseñada para gestionar integralmente una flota de vehículos. El propósito fundamental es establecer una herramienta central que no solo garantice la **integridad, seguridad y disponibilidad de los datos**, sino que también facilite el acceso controlado y específico a diversos actores clave: personal administrativo, centros de servicio técnico, operadores de vehículos y dueños de los activos.

El proceso se adhirió a una metodología rigurosa: desde la recopilación de requerimientos operacionales, la elaboración de un **Modelo Entidad-Relación (MER)**, la formalización de un esquema lógico **normalizado**, hasta su despliegue

físico. Adicionalmente, se incorporó un robusto esquema de seguridad que utiliza **vistas SQL** para lograr una "segmentación virtual" de los datos, asegurando que cada rol acceda exclusivamente a la información pertinente a sus responsabilidades.

2. Fundamentos Teóricos

2.1. El Modelo Relacional y su Aplicación

El modelo de datos relacional, basado en la teoría de conjuntos, organiza la información en tablas conectadas mediante atributos comunes. Esta arquitectura es fundamental para el proyecto debido a su capacidad para garantizar la **consistencia** y la facilidad en la ejecución de **consultas complejas** a través del lenguaje SQL, minimizando la redundancia de datos.

2.2. Plataforma de Implementación: MySQL

Se seleccionó MySQL, un sistema de código abierto, reconocido por su **alto rendimiento, escalabilidad y compatibilidad** con el estándar SQL. Su uso permite una implementación robusta que asegura las propiedades de **atomicidad, consistencia, aislamiento y durabilidad (ACID)** para todas las transacciones de gestión vehicular.

2.3. Estructuración del Diseño de la Base de Datos

El diseño se desarrolló en tres etapas fundamentales:

1. **Conceptual:** Definición de los principales elementos (entidades), sus características (atributos) y los vínculos entre ellos, resultando en un MER independiente de la tecnología.
2. **Lógica:** Conversión del MER en un esquema de tablas, definiendo **claves primarias (PK)** y **claves foráneas (FK)**, y aplicando el proceso de normalización.
3. **Física:** Decisiones específicas del RDBMS (MySQL) para la optimización de desempeño, como la selección de tipos de datos e indexación.

2.4. Normalización de Datos

La normalización se aplicó para estructurar las tablas, reduciendo la redundancia y mejorando la integridad. Se implementó un diseño que cumple con la **Tercera Forma Normal (3NF)**, asegurando que la dependencia de cada campo sea exclusiva de la clave primaria de su tabla.

2.5. Seguridad y Control de Acceso Mediante Vistas

Para implementar el control de acceso, se empleó la técnica de **fragmentación virtual** a través de **Vistas SQL**. Una vista funciona como una tabla virtual, permitiendo exponer solo un subconjunto específico de datos a un usuario o rol sin concederle permisos directos sobre las tablas base. Esta estrategia asegura el **principio del mínimo privilegio**.

3. Metodología de Desarrollo

3.1. Requisitos Operacionales

El análisis funcional se centró en las siguientes necesidades de gestión:

- **Entidades Centrales:** Vehículo, Conductor, Dueño del Activo, Taller de Mantenimiento, Registro de Ruta, Transacciones y Documentos Legales.
- **Funcionalidades Clave:** Seguimiento detallado de egresos (peajes, combustible), monitoreo de vencimientos (licencias, seguros), y registro integral de servicios (preventivos y correctivos).
- **Perfiles de Acceso (Roles):**
 - **Administrativo:** Vista global para control de costos, estado de toda la flota y alertas.
 - **Taller:** Acceso restringido al historial técnico de las unidades que atienden.
 - **Conductor:** Consulta de itinerarios asignados y registro de gastos operacionales.
 - **Dueño:** Visibilidad sobre el rendimiento, costos de servicio y estado legal de sus activos.

3.2. Estructura Lógica Normalizada

El modelo conceptual se tradujo en el siguiente esquema relacional, con énfasis en la 3NF:

- **Dueño Activo** (idDueño, nombre, rfc, contacto)
- **Unidad Móvil** (idUnidad, placa, marca, modelo, año, nSerie, idDueño)
- **Personal Operativo** (idConductor, nombre, licencia, vencimientoLicencia)
- **Centro Servicio** (idTaller, nombre, dirección, teléfono)
- **Servicio Mantenimiento** (idMantenimiento, idUnidad, idTaller, fechaServicio, detalle, costo)
- **Documentacion** (idDoc, idUnidad, tipoDoc, vencimiento)
- **Transaccion** (idGasto, idUnidad, idConductor, concepto, monto, fecha)
- **Ruta Asignada** (idRuta, idUnidad, idConductor, origen, destino, fechaInicio, estado)

3.3. Implementación de la Seguridad

La seguridad se implementó a través de Vistas para la segmentación de datos. Solo estas vistas reciben los permisos de consulta (GRANT SELECT) para cada rol, garantizando la privacidad y el control de la información sensible.

4. Desarrollo Físico (MySQL)

4.1. Definición del Esquema (DDL - Fragmento)

El código SQL DDL creó la estructura base y las claves foráneas que garantizan la integridad referencial.

SQL

```
CREATE DATABASE IF NOT EXISTS control_logistico;
```

```
USE control_logistico;
```

```
CREATE TABLE Dueño_Activo (  
    idDueño INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100) NOT NULL,  
    rfc VARCHAR(13) UNIQUE  
);
```

```
CREATE TABLE Unidad_Movil (  
    idUnidad INT PRIMARY KEY AUTO_INCREMENT,  
    placa VARCHAR(10) UNIQUE NOT NULL,  
    marca VARCHAR(50),  
    modelo VARCHAR(50),  
    anio INT,  
    nSerie VARCHAR(20) UNIQUE,  
    idDueño INT,  
    FOREIGN KEY (idDueño) REFERENCES Dueño_Activo(idDueño)  
);
```

-- ... se crearon las tablas restantes con sus respectivas FKs.

4.2. Vistas de Acceso por Rol (Fragmentos)

Vista para el Rol Administrativo

Diseñada para una supervisión integral, combinando información de activos, personal y estado operacional.

SQL

```
CREATE OR REPLACE VIEW V_Supervisor_Flota AS
```

```
SELECT
```

```
    um.placa, um.marca, um.modelo,
```

```
    po.nombre AS conductor_asignado,
```

```
    ra.origen, ra.destino, ra.estado AS estado_ruta,
```

```
    sm.fechaServicio
```

```
FROM Unidad_Movil um
```

```
LEFT JOIN Ruta_Asignada ra ON um.idUnidad = ra.idUnidad
```

```
LEFT JOIN Personal_Operativo po ON ra.idConductor = po.idConductor
```

```
LEFT JOIN Servicio_Mantenimiento sm ON um.idUnidad = sm.idUnidad;
```

-- Ejemplo de uso:

```
SELECT placa, estado_ruta FROM V_Supervisor_Flota WHERE estado_ruta =  
'Activa';
```

Vista para el Centro de Servicio (Taller)

Restringida a los datos técnicos y el historial de reparaciones para un taller específico (ej. ID 5).

SQL

```
CREATE OR REPLACE VIEW V_Servicio_Tecnico_5 AS
```

```
SELECT
```

```
    um.placa, um.marca, um.modelo, um.anio, um.nSerie,
```

```
    sm.fechaServicio, sm.detalle, sm.costo
```

```
FROM Servicio_Mantenimiento sm  
JOIN Unidad_Movil um ON sm.idUnidad = um.idUnidad  
WHERE sm.idTaller = 5;
```

-- Ejemplo de uso:

```
SELECT placa, costo FROM V_Servicio_Tecnico_5 ORDER BY costo DESC;
```

5. Conclusiones y Proyecciones

Se culminó exitosamente la implementación de un sistema de base de datos relacional en MySQL, satisfaciendo los requerimientos para la gestión integral de activos vehiculares. La combinación del **modelo de datos normalizado (3NF)** y la arquitectura de **seguridad basada en Vistas SQL** asegura un ambiente de datos **íntegro, escalable y con acceso estrictamente controlado**.

Esta plataforma centraliza la información crucial, permitiendo a la dirección tomar **decisiones informadas**, optimiza los flujos de trabajo de los operadores y talleres, y ofrece la transparencia necesaria a los dueños de los activos.

Trabajo Futuro: El siguiente paso recomendado es el desarrollo de una capa de interfaz de usuario (aplicación web o móvil) para facilitar la captura de datos por parte de los operadores en campo y la presentación de análisis y **tableros de control (dashboards)** intuitivos para la alta gerencia.

Bibliografía

Edenred México. (2023). Flotilla de autos: ¿cómo administrarla de forma eficiente?.

MySQL Documentation. (2025). MySQL 8.0 Reference Manual. Oracle Corporation.