

**AutoRental**

**Daniel Alejandro Latorre Ruiz**  
**CampusLands**  
**P1**

**Pedro Gomez Bonilla**  
**25/06/2024**

## Tabla de contenidos

<b>Introducción</b>	<b>5</b>
<b>Caso de Estudio</b>	<b>6</b>
<b>Instalación General</b>	<b>6</b>
<b>Planificación</b>	<b>7</b>
<b>Ejecución</b>	<b>8</b>
MODELO CONCEPTUAL	8
Descripción	8
Gráfica	8
Descripción Técnica	9
• Sucursales:	9
• Empleados:	9
• Clientes:	9
• Vehículos:	9
• Alquileres:	9
MODELO LÓGICO	9
Descripción	9
Gráfica	10
Descripción Técnica	10
Relaciones entre entidades	10
NORMALIZACIÓN	10
Primera Forma Normal (1FN)	10
Descripción	10
Descripción Técnica	11
Segunda Forma Normal (2FN)	11
Descripción	11
Descripción Técnica	11
Tercera Forma Normal (3FN)	11
Descripción	11
Descripción Técnica	11
MODELO FÍSICO	11
Descripción	11
Gráficas	12
Descripción Técnica	13
DIAGRAMA UML	13
Descripción	13
Gráfica	13
Descripción Técnica	14
FUNCIONES Y PROCEDIMIENTOS	15
Funciones	16
Procedimientos	16
1. Procedimiento 1()	16

2. Procedimiento 2()	16
3. Procedimiento 3()	18
USUARIOS Y ACCESOS	18
<b>REFERENCIAS</b>	<b>19</b>

# Introducción

Desarrollo de un sistema de información para AutoRental, una empresa dedicada al alquiler de vehículos. La compañía cuenta actualmente con cinco sucursales en diferentes ciudades y planea expandirse a otras localidades del país. AutoRental posee una flota diversa de vehículos, que incluye distintos tipos, modelos, capacidades y características.

El sistema de información debe gestionar el alquiler de vehículos, permitiendo que los clientes puedan alquilar un vehículo en una sucursal y devolverlo en otra. Además, AutoRental ofrece descuentos en varios tipos de vehículos durante todo el año, y los precios de alquiler varían según el tipo de vehículo, cobrando tarifas diarias y semanales. En caso de que un cliente entregue el vehículo después de la fecha acordada, se aplicará un recargo del 8% por los días adicionales.

La base de datos debe almacenar información detallada sobre las sucursales, empleados, clientes, vehículos y alquileres. Asimismo, se requiere el desarrollo de tres aplicaciones de software: una herramienta de gestión interna para los empleados de AutoRental, una aplicación web para los clientes y una aplicación para Android, ambas con funcionalidades como registro, inicio de sesión, consulta de disponibilidad de vehículos, alquiler y consulta de historial de alquileres.

El objetivo es implementar la lógica de negocio para las operaciones clave en los aplicativos de software para los clientes, garantizando que se cubran todas las necesidades operativas y de gestión de AutoRental.

# Caso de Estudio

La empresa AutoRental busca un sistema de información de alquiler de vehículos, esta cuenta con 5 sucursales en diferentes ciudades y se proyecta a expandirse a otras ciudades del país y cuenta con una flota propia de vehículos de diferentes tipos, modelos (año), capacidad, etc. Los clientes de AutoRental podrán alquilar un vehículo en una sucursal y entregarlo en otra sucursal.

AutoRental ofrece descuentos sobre diferentes tipos de vehículos a lo largo del año. Los valores de alquiler dependen del tipo de vehículo (sedán, compacto, camioneta platón, camioneta lujo, deportivo, etc) y se cobran por días y/o semanas. Por ejemplo, si un alquila un vehículo por 9 días, el valor cotizado será de 1 semana y 2 días.

Si un cliente entrega el vehículo pasada la fecha de entrega contratada, se cobrarán los días adicionales con un incremento del 8%.

## Instalación General

Los archivos presentados en el [repositorio](#) cuentan con todo el trabajo de este sistema de información.

Este proyecto se divide en 4 partes:

- [Tablas](#): Aquí puedes encontrar las 5 entidades y los atributos de cada una con los cuales se va a trabajar.
- [Inserciones](#): En este apartado se añade la información a cada atributo creado en las tablas anteriores.
- [Permisos](#): Este posee los permisos que tienen los 3 diferentes usuarios Cliente, Empleado y Jefe.
- [Funciones](#): Estas son las funciones las cuales se le asignan a los usuarios dependiendo de su rol.

# Planificación

Primero, se desarrollará el **modelo conceptual** del sistema de alquiler de vehículos. Este modelo incluirá la identificación de las entidades principales, como Sucursales, Empleados, Clientes, Vehículos y Alquileres, así como las relaciones entre ellas. Se definirán los atributos clave para cada entidad para asegurar una representación clara y completa del dominio del problema.

A continuación, se realizará la **conversión del modelo conceptual al modelo lógico**. En esta fase, se especificarán las claves primarias y foráneas para cada entidad y relación, garantizando la integridad referencial mediante la definición de restricciones adecuadas. Este paso es crucial para asegurar que los datos sean consistentes y se mantengan relaciones precisas entre las tablas.

Posteriormente, se procederá con la **normalización del modelo lógico** en tres niveles. Primero, se aplicará la Primera Forma Normal (1NF), asegurando que todos los atributos contengan valores atómicos y que cada registro sea único. Después, se aplicará la Segunda Forma Normal (2NF), eliminando dependencias parciales para asegurar que todos los atributos no clave dependan completamente de la clave primaria. Finalmente, se aplicará la Tercera Forma Normal (3NF), eliminando dependencias transitivas para que los atributos no clave dependan únicamente de la clave primaria.

Una vez normalizado el modelo lógico, se traducirá al **modelo físico**. En esta fase, se definirá el esquema de la base de datos, incluyendo la creación de tablas, índices y restricciones necesarias. También se optimizarán los índices y consultas para asegurar la eficiencia y el rendimiento del sistema de base de datos.

A continuación, se elaborará el **diagrama UML** para visualizar la estructura y las relaciones entre las entidades del sistema. Este diagrama incluirá diagramas de clases, relaciones y cardinalidades, proporcionando una representación gráfica detallada del diseño del sistema y facilitando la comprensión de su arquitectura.

Seguidamente, se desarrollarán los **procedimientos almacenados**, funciones y triggers necesarios para el funcionamiento del sistema. Estos procedimientos deben cumplir con los requisitos funcionales y de seguridad del sistema, asegurando que las operaciones se realicen de manera eficiente y segura.

Finalmente, se realizará la **gestión de usuarios y roles** en el sistema de base de datos. Se crearán usuarios con roles específicos: Cliente, Empleado y Jefe. Se asignarán permisos adecuados a cada usuario basado en su rol, garantizando el principio de menor privilegio. Se implementarán controles de acceso y seguridad para proteger los datos sensibles y asegurar que cada usuario solo tenga acceso a la información y funcionalidades necesarias para su rol.

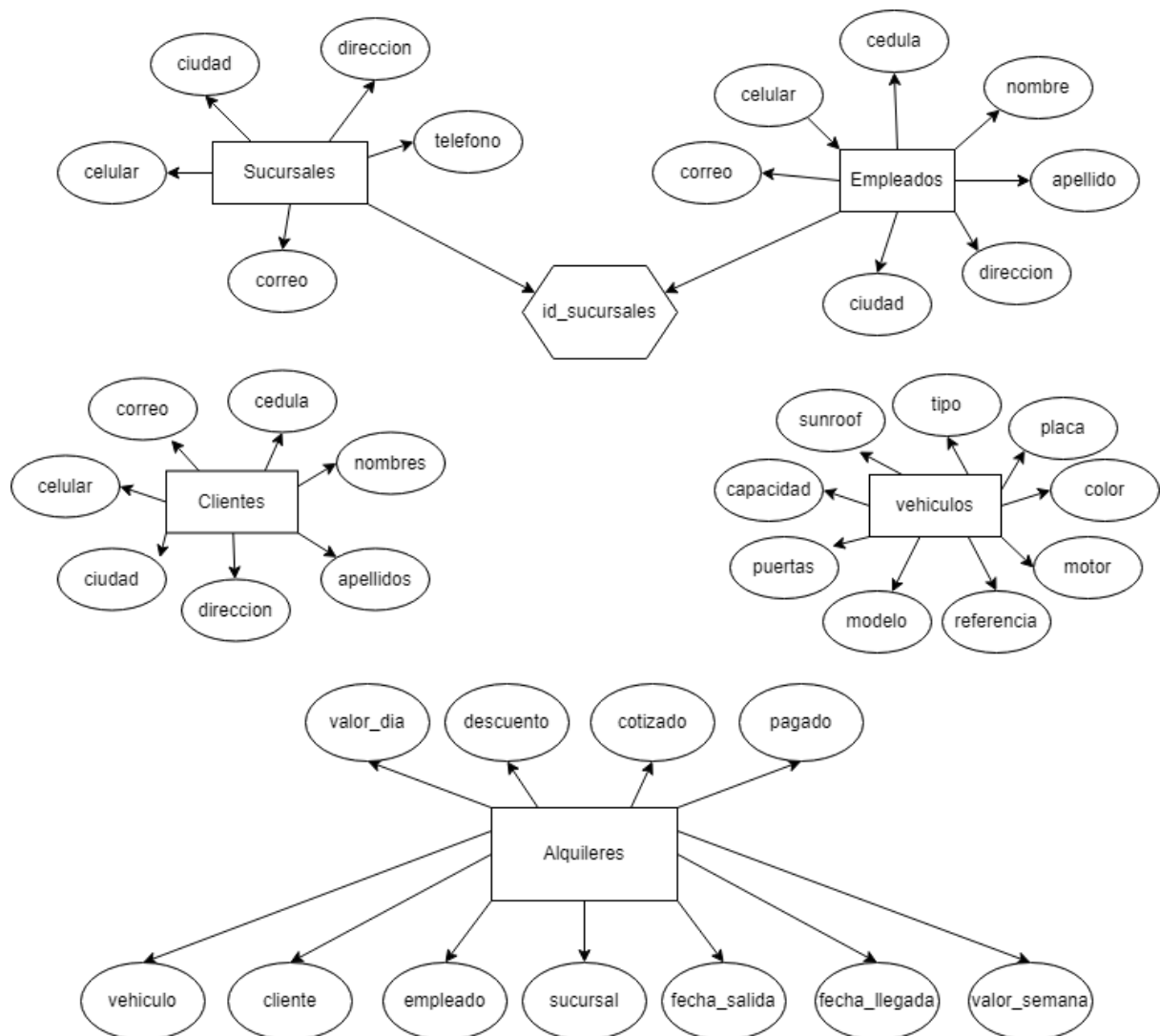
# Ejecución

## MODELO CONCEPTUAL

### Descripción

Se realizó el modelo conceptual de la base de datos AutoRental la cual contiene 5 entidades y cada una con sus diferentes atributos las cuales trabajan entre sí.

### Gráfica



## Descripción Técnica

La base de datos deberá cumplir almacenar la siguiente información:

- Sucursales:

ciudad y dirección donde se ubica, teléfono fijo, celular y correo electrónico.

- Empleados:

sucursal donde labora, cédula, nombres, apellidos, dirección y ciudad de residencia, celular y correo electrónico.

- Clientes:

cédula, nombres, apellidos, dirección y ciudad de residencia, celular y correo electrónico.

- Vehículos:

tipo de vehículo, placa, referencia, modelo, puertas, capacidad, sunroof, motor, color.

- Alquileres:

vehículo, cliente, empleado, sucursal y fecha de salida, sucursal y fecha de llegada, fecha esperada de llegada, valor de alquiler por semana, valor de alquiler por día, porcentaje de descuento, valor cotizado y valor pagado.

## MODELO LÓGICO

### Descripción

Se realizó la conversión del modelo conceptual al modelo lógico, agregando las tablas correspondientes junto con sus atributos. En este proceso, se destacaron las llaves primarias (PK) y las llaves foráneas (FK) para cada tabla, asegurando la integridad referencial y las relaciones adecuadas entre las entidades.



## Gráfica

Sucursales			Empleados			Alquileres	
PK	id		PK	id		PK	id_alquiler
	ciudad			cedulas		FK	vehiculo_id
	direccion			nombre		FK	cliente_id
	telefono			apellido		FK	empleado_id
	correo			correo		FK	sucursal_id
				celular			fecha_salida
				ciudad			fecha_llegada
				direccion			valor_dia
							valor_semana
Cientes							descuento
PK	id		Vehiculos				cotizado
	cedula		PK	id			pagado
	nombres			placa			
	apellidos			tipo			
	correo			color			
	celular			motor			
	ciudad			modelo			
	direccion			referencia			
				capacidad			
				puertas			
				sunroof			

## Descripción Técnica

### Relaciones entre entidades

1. **Sucursales** tiene una relación de uno a muchos con **Alquileres**.
2. **Cientes** tiene una relación de uno a muchos con **Alquileres**.
3. **Empleados** tiene una relación de uno a muchos con **Alquileres**.
4. **Vehículos** tiene una relación de uno a muchos con **Alquileres**.

## NORMALIZACIÓN

### Primera Forma Normal (1FN)

#### Descripción

En la 1NF, cada columna debe contener valores atómicos, y cada registro debe ser único.

#### Descripción Técnica

- **Valor Atómico:** Cada valor en una columna debe ser indivisible.
- **Clave Primaria:** Debe haber una clave primaria que identifique de manera única cada fila.

## Segunda Forma Normal (2FN)

### Descripción

En la 2NF, la tabla debe estar en 1NF y todos los atributos no clave deben depender completamente de la clave primaria.

### Descripción Técnica

- **Elimina Dependencias Parciales:** Atributos no clave deben depender completamente de la clave primaria y no solo de una parte de una clave compuesta.

## Tercera Forma Normal (3FN)

### Descripción

En la 3NF, la tabla debe estar en 2NF y no debe haber dependencias transitivas entre los atributos no clave.

### Descripción Técnica

- **Elimina Dependencias Transitivas:** Ningún atributo no clave debe depender de otro atributo no clave; todos los atributos no clave deben depender únicamente de la clave primaria.

## MODELO FÍSICO

### Descripción

Se avanzó con la implementación del modelo físico utilizando las tablas definidas previamente. En esta etapa, se creó la base de datos denominada "autoRental" y, a continuación, se procedió a la creación de las tablas correspondientes a cada entidad. Todo este proceso se llevó a cabo utilizando la herramienta MySQL Workbench.

## Gráficas

```
CREATE DATABASE autoRental;  
use autoRental;
```

```
-- Tabla Sucursales
```

```
CREATE TABLE Sucursales (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    ciudad VARCHAR(50),  
    direccion VARCHAR(100),  
    telefono VARCHAR(20),  
    correo VARCHAR(50)  
);
```

```
-- Tabla Clientes
```

```
CREATE TABLE Clientes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    cedula VARCHAR(20),  
    nombres VARCHAR(50),  
    apellidos VARCHAR(50),  
    correo VARCHAR(50),  
    celular VARCHAR(20),  
    ciudad VARCHAR(50),  
    direccion VARCHAR(100)  
);
```

```
-- Tabla Empleados
```

```
CREATE TABLE Empleados (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    cedula VARCHAR(20),  
    nombre VARCHAR(50),  
    apellido VARCHAR(50),  
    correo VARCHAR(50),  
    celular VARCHAR(20),  
    ciudad VARCHAR(50),  
    direccion VARCHAR(100)  
);
```

```
-- Tabla Vehiculos
```

```
CREATE TABLE Vehiculos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    placa VARCHAR(20),  
    tipo VARCHAR(20),  
    color VARCHAR(20),  
    motor VARCHAR(20),  
    modelo VARCHAR(20),  
    referencia VARCHAR(20),  
    capacidad INT,  
    puertas INT,  
    sunroof BOOLEAN  
);
```

```
-- Tabla Alquileres
```

```
CREATE TABLE Alquileres (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    vehiculo_id INT,  
    cliente_id INT,  
    empleado_id INT,  
    sucursal_id INT,  
    fecha_salida DATE,  
    fecha_llegada DATE,  
    valor_dia DECIMAL(10, 2),  
    valor_semana DECIMAL(10, 2),  
    descuento DECIMAL(10, 2),  
    cotizado BOOLEAN,  
    pagado BOOLEAN,  
    FOREIGN KEY (vehiculo_id) REFERENCES Vehiculos(id),  
    FOREIGN KEY (cliente_id) REFERENCES Clientes(id),  
    FOREIGN KEY (empleado_id) REFERENCES Empleados(id),  
    FOREIGN KEY (sucursal_id) REFERENCES Sucursales(id)  
);
```

## Descripción Técnica

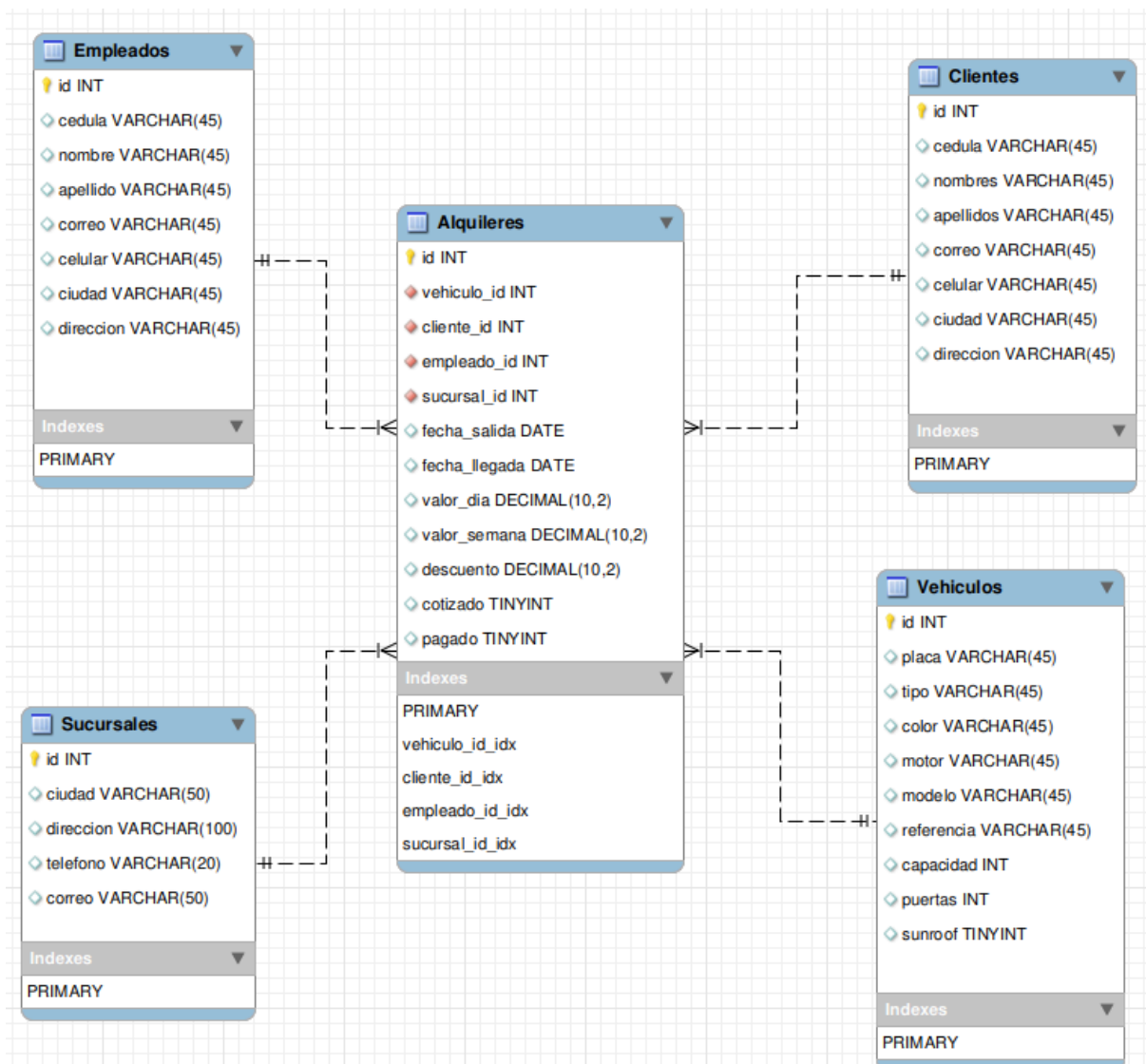
Después de esto se continuó con las inserciones de cada tabla para manejar información, en la tabla sucursales se hicieron 5 inserciones y en las demás tablas se hicieron 100 inserciones, rellendo cada atributo de las tablas, esto se hace con el fin de que más adelante se le dará uso mediante consultas.

## DIAGRAMA UML

### Descripción

Se realizó el diagrama de la base de datos AutoRental teniendo en cuenta las 5 entidades con sus atributos con sus relaciones, todo esto se trabajó utilizando la herramienta MySQL-Workbench.

### Gráfica



# Descripción Técnica

## Entidades y Atributos

### 1. Empleados

- **id (INT)**: Identificador único del empleado (clave primaria).
- **cedula (VARCHAR(45))**: Número de cédula del empleado.
- **nombre (VARCHAR(45))**: Nombre del empleado.
- **apellido (VARCHAR(45))**: Apellido del empleado.
- **direccion (VARCHAR(45))**: Dirección del empleado.
- **ciudad (VARCHAR(45))**: Ciudad de residencia del empleado.
- **celular (VARCHAR(45))**: Número de celular del empleado.
- **correo (VARCHAR(45))**: Correo electrónico del empleado.

### 2. Clientes

- **id (INT)**: Identificador único del cliente (clave primaria).
- **cedula (VARCHAR(45))**: Número de cédula del cliente.
- **nombres (VARCHAR(45))**: Nombres del cliente.
- **apellidos (VARCHAR(45))**: Apellidos del cliente.
- **correo (VARCHAR(45))**: Correo electrónico del cliente.
- **celular (VARCHAR(45))**: Número de celular del cliente.
- **ciudad (VARCHAR(45))**: Ciudad de residencia del cliente.
- **direccion (VARCHAR(45))**: Dirección del cliente.

### 3. Sucursales

- **id (INT)**: Identificador único de la sucursal (clave primaria).
- **ciudad (VARCHAR(50))**: Ciudad donde se ubica la sucursal.
- **direccion (VARCHAR(45))**: Dirección de la sucursal.
- **telefono (VARCHAR(45))**: Número de teléfono fijo de la sucursal.
- **celular (VARCHAR(45))**: Número de celular de la sucursal.
- **correo (VARCHAR(45))**: Correo electrónico de la sucursal.

### 4. Vehiculos

- **id (INT)**: Identificador único del vehículo (clave primaria).
- **placa (VARCHAR(45))**: Número de placa del vehículo.
- **tipo (VARCHAR(45))**: Tipo de vehículo.
- **color (VARCHAR(45))**: Color del vehículo.
- **motor (VARCHAR(45))**: Número de motor del vehículo.
- **modelo (VARCHAR(45))**: Modelo del vehículo.
- **referencia (VARCHAR(45))**: Referencia del vehículo.
- **capacidad (INT)**: Capacidad del vehículo.
- **puertas (INT)**: Número de puertas del vehículo.
- **sunroof (TINYINT)**: Indicador de si el vehículo tiene sunroof (1 para sí, 0 para no).

### 5. Alquileres

- **id (INT)**: Identificador único del alquiler (clave primaria).
- **vehiculo\_id (INT)**: Identificador del vehículo alquilado (clave foránea que referencia a Vehiculos.id).
- **cliente\_id (INT)**: Identificador del cliente que realiza el alquiler (clave foránea que referencia a Clientes.id).

- **empleado\_id (INT)**: Identificador del empleado que gestiona el alquiler (clave foránea que referencia a Empleados.id).
- **sucursal\_id (INT)**: Identificador de la sucursal donde se realiza el alquiler (clave foránea que referencia a Sucursales.id).
- **fecha\_salida (DATE)**: Fecha de salida del vehículo.
- **fecha\_llegada (DATE)**: Fecha de llegada del vehículo.
- **valor\_dia (DECIMAL(10,2))**: Valor del alquiler por día.
- **valor\_semana (DECIMAL(10,2))**: Valor del alquiler por semana.
- **descuento (DECIMAL(10,2))**: Porcentaje de descuento aplicado.
- **cotizado (TINYINT)**: Indicador de si el alquiler fue cotizado (1 para sí, 0 para no).
- **pagado (TINYINT)**: Indicador de si el alquiler fue pagado (1 para sí, 0 para no).

## Relaciones

- Alquileres - Vehiculos
  - Relación de muchos a uno: Cada alquiler está asociado con un solo vehículo, pero un vehículo puede estar asociado con muchos alquileres.
  - Clave foránea: vehiculo\_id en Alquileres referencia id en Vehiculos.
- Alquileres - Clientes
  - Relación de muchos a uno: Cada alquiler está asociado con un solo cliente, pero un cliente puede tener muchos alquileres.
  - Clave foránea: cliente\_id en Alquileres referencia id en Clientes.
- Alquileres - Empleados
  - Relación de muchos a uno: Cada alquiler está asociado con un solo empleado, pero un empleado puede gestionar muchos alquileres.
  - Clave foránea: empleado\_id en Alquileres referencia id en Empleados.
- Alquileres - Sucursales
  - Relación de muchos a uno: Cada alquiler está asociado con una sola sucursal, pero una sucursal puede gestionar muchos alquileres.
  - Clave foránea: sucursal\_id en Alquileres referencia id en Sucursales.

## Claves y Restricciones

- **Claves Primarias (PRIMARY KEY)**: Cada tabla tiene una clave primaria (**id**) que garantiza la unicidad de cada registro.
- **Claves Foráneas (FOREIGN KEY)**: Las claves foráneas en Alquileres (vehiculo\_id, cliente\_id, empleado\_id, sucursal\_id) aseguran la integridad referencial entre las tablas relacionadas.

## FUNCIONES Y PROCEDIMIENTOS

## Funciones

## Procedimientos

### 1. Procedimiento 1()

Se empezó creando los procedimientos para el rol de clientes los cuales son:

- Consultar los vehículos.
- Consultar el historial de alquileres.

```
-- Clientes
```

```
-- Procedimiento para consultar todos los vehículos disponibles
```

```
DELIMITER //
```

```
CREATE PROCEDURE consultarVehiculos ()
```

```
⌋ BEGIN
```

```
    SELECT * FROM Vehiculos;
```

```
⌋ END //
```

```
DELIMITER ;
```

```
-- Procedimiento para consultar el historial de alquileres de un cliente específico
```

```
DELIMITER //
```

```
⌋ CREATE PROCEDURE consultarHistorialAlquileres (
```

```
    IN p_cedula_cliente VARCHAR(20)
```

```
⌋ )
```

```
⌋ BEGIN
```

```
    SELECT *
```

```
    FROM Alquileres
```

```
    WHERE cliente = p_cedula_cliente;
```

```
⌋ END //
```

```
DELIMITER ;
```

### 2. Procedimiento 2()

El rol de empleados posee lo siguiente:

- Crear un vehículo nuevo.
- Eliminar un vehículo.
- Actualizar un vehículo ya existente.

```

-- Empleados

-- Procedimiento para crear un nuevo vehículo
DELIMITER //
CREATE PROCEDURE crearVehiculo (
    IN p_placa VARCHAR(20),
    IN p_tipo VARCHAR(20),
    IN p_color VARCHAR(20),
    IN p_motor VARCHAR(20),
    IN p_modelo VARCHAR(20),
    IN p_referencia VARCHAR(20),
    IN p_capacidad INT,
    IN p_puertas INT,
    IN p_sunroof BOOLEAN
)
BEGIN
    INSERT INTO Vehiculos (placa, tipo, color, motor, modelo, referencia, capacidad, puertas, sunroof)
    VALUES (p_placa, p_tipo, p_color, p_motor, p_modelo, p_referencia, p_capacidad, p_puertas, p_sunroof);
END //
DELIMITER ;

-- Procedimiento para actualizar un vehículo existente
DELIMITER //
CREATE PROCEDURE actualizarVehiculo (
    IN p_placa VARCHAR(20),
    IN p_tipo VARCHAR(20),
    IN p_color VARCHAR(20),
    IN p_motor VARCHAR(20),
    IN p_modelo VARCHAR(20),
    IN p_referencia VARCHAR(20),
    IN p_capacidad INT,
    IN p_puertas INT,
    IN p_sunroof BOOLEAN
)
BEGIN
    UPDATE Vehiculos
    SET tipo = p_tipo,
        color = p_color,
        motor = p_motor,
        modelo = p_modelo,
        referencia = p_referencia,
        capacidad = p_capacidad,
        puertas = p_puertas,
        sunroof = p_sunroof
    WHERE placa = p_placa;
END //
DELIMITER ;

```



```

-- Procedimiento para eliminar un vehículo
DELIMITER //
CREATE PROCEDURE eliminarVehiculo (
    IN p_placa VARCHAR(20)
)
BEGIN
    DELETE FROM Vehiculos
    WHERE placa = p_placa;
END //
DELIMITER ;

```

### 3. Procedimiento 3()

Terminando con el rol de jefe el cual tiene acceso a esto:

- Consultar todas las tablas de la base de datos.

```

-- Jefe

-- Procedimiento para consultar todos los datos de las tablas
DELIMITER //
CREATE PROCEDURE consultarTodo ()
BEGIN
    SELECT * FROM Vehiculos;
    SELECT * FROM Clientes;
    SELECT * FROM Empleados;
    SELECT * FROM Sucursales;
    SELECT * FROM Alquileres;
END //
DELIMITER ;

```

## USUARIOS Y ACCESOS

Se crearon los procedimientos para después asignarle estos a cada usuario dependiendo de su rol (Cliente, Empleado y Jefe).

```

CREATE USER 'Empleados'@'localhost' IDENTIFIED BY 'contraseña';
CREATE USER 'Clientes'@'localhost' IDENTIFIED BY 'contraseña';
CREATE USER 'Jefe'@'localhost' IDENTIFIED BY 'contraseña';

```

Terminando este proceso se le asignó a cada usuario los procedimientos dependiendo de su rol utilizando GRANT para esto.

```
-- Asignar permisos a cada usuario según sus funciones
GRANT EXECUTE ON PROCEDURE crearVehiculo TO 'Empleados'@'localhost';
GRANT EXECUTE ON PROCEDURE actualizarVehiculo TO 'Empleados'@'localhost';
GRANT EXECUTE ON PROCEDURE eliminarVehiculo TO 'Empleados'@'localhost';

GRANT EXECUTE ON PROCEDURE consultarVehiculos TO 'Clientes'@'localhost';
GRANT EXECUTE ON PROCEDURE consultarHistorialAlquileres TO 'Clientes'@'localhost';

GRANT EXECUTE ON PROCEDURE consultarTodo TO 'Jefe'@'localhost';
```

## REFERENCIAS