

# Informe Ejercicio Dia4

Deivid Velasquez Gutierrez

CampusLand, MySQL II

Pedro Gomez

19 de junio del 2024

## Modelo Conceptual:

Se realizará el modelo conceptual con sus relaciones, entidades y atributos de cada entidad con el fin de poder interpretar mejor la parte. (El modelo conceptual se encuentra en formato jpg en el repositorio como ModeloConceptual).

## Modelo Lógico:

El modelo lógico nos va a permitir observar qué contenido va a tener cada tabla y de qué tipo será este atributo con el fin de entender qué información vamos a tener de cada tabla.

- sucursal(
  - id int AUTO\_INCREMENT PRIMARY KEY,
  - ciudad VARCHAR(100) NOT NULL,
  - direccion VARCHAR(200) NOT NULL,
  - telefono\_fijo VARCHAR(20),
  - celular VARCHAR(20),
  - correo\_electronico VARCHAR(100)
 );
- empleado(
  - id INT AUTO\_INCREMENT PRIMARY KEY,
  - sucursal\_id INT,
  - Foreign Key (sucursal\_id) REFERENCES sucursal(id),
  - cedula VARCHAR(20) UNIQUE NOT NULL,
  - nombre1 VARCHAR(100) NOT NULL,
  - nombre2 VARCHAR(100),
  - apellido1 VARCHAR(100) NOT NULL,
  - apellido2 VARCHAR(100),
  - direccion VARCHAR(200) NOT NULL,
  - ciudad\_residencia VARCHAR(100) NOT NULL,
  - celular VARCHAR(20),
  - correo\_electronico VARCHAR(100)
 );
- cliente (
  - id INT AUTO\_INCREMENT PRIMARY KEY,
  - cedula VARCHAR(20) UNIQUE NOT NULL,
  - nombre1 VARCHAR(100) NOT NULL,
  - nombre2 VARCHAR(100),
  - apellido1 VARCHAR(100) NOT NULL,
  - apellido2 VARCHAR(100),
  - direccion VARCHAR(200) NOT NULL,
  - ciudad\_residencia VARCHAR(100) NOT NULL,
  - celular VARCHAR(20),
  - correo\_electronico VARCHAR(100)
 );

- vehiculo(
  - id INT AUTO\_INCREMENT PRIMARY KEY,
  - tipo ENUM(('sedán', 'compacto', 'camioneta platón', 'camioneta lujo', 'deportivo'),
  - placa VARCHAR(20) UNIQUE NOT NULL,
  - referencia VARCHAR(100),
  - modelo INT NOT NULL,
  - puertas INT NOT NULL,
  - capacidad INT NOT NULL,
  - sunroof BOOLEAN,
  - motor VARCHAR(50),
  - color VARCHAR(30)
 );
  
- alquileres(
  - id INT AUTO\_INCREMENT PRIMARY KEY,
  - vehiculo\_id INT NOT NULL,
  - Foreign Key (vehiculo\_id) REFERENCES vehiculo(id),
  - cliente\_id INT NOT NULL,
  - Foreign Key (cliente\_id) REFERENCES cliente(id),
  - empleado\_id INT NOT NULL,
  - Foreign Key (empleado\_id) REFERENCES empleado(id),
  - sucursal\_salida\_id INT NOT NULL,
  - Foreign Key (sucursal\_salida\_id) REFERENCES sucursal(id),
  - sucursal\_llegada\_id INT NOT NULL,
  - Foreign Key (sucursal\_llegada\_id) REFERENCES sucursal(id),
  - fecha\_salida DATE NOT NULL,
  - fecha\_llegada DATE NOT NULL,
  - fecha\_esperada\_llegada DATE NOT NULL,
  - valor\_alquiler\_semana DECIMAL(10, 2) NOT NULL,
  - valor\_alquiler\_dia DECIMAL(10, 2) NOT NULL,
  - porcentaje\_descuento DECIMAL(5, 2),
  - valor\_cotizado DECIMAL(10, 2) NOT NULL,
  - valor\_pagado DECIMAL(10, 2)
 );

### Relaciones:

El modelo de relaciones lo encuentras en el repositorio en el formato png como ModeloLogico.

### Normalización:

La normalización se realizó en el transcurso del proyecto optimizando lo mejor posible los datos cumpliendo con cada uno de sus objetivos principales.

## Modelo Físico:

Este modelo se desarrolló en MySQL, utilizando las tablas ya normalizadas y con sus atributos correspondientes. Por último se Insertaron 100 datos por cada tabla meno en la de sucursal, ya que está en requerimientos nos presenta que cuenta solo con 5 sucursales en diferentes ciudades.

## Procedimientos:

Se implementaron una serie de procedimiento los cuales permiten al usuario ya sea empleado o cliente realizar unas funciones diferentes dentro de las tablas. A continuación verás una explicación breve de la función que cumple cada procedimiento:

### Registro de Empleado

**Nombre del Procedimiento:** registro\_empleado

**Descripción:** Inserta un nuevo registro de empleado en la tabla empleado, siempre que el id de la sucursal exista.

#### Parámetros:

- empleado\_id\_sucursal INT: ID de la sucursal del empleado.
- empleado\_cedula VARCHAR(20): Cédula del empleado.
- empleado\_nombre1 VARCHAR(100): Primer nombre del empleado.
- empleado\_nombre2 VARCHAR(100): Segundo nombre del empleado (puede ser NULL).
- empleado\_apellido1 VARCHAR(100): Primer apellido del empleado.
- empleado\_apellido2 VARCHAR(100): Segundo apellido del empleado (puede ser NULL).
- empleado\_direccion VARCHAR(200): Dirección del empleado.
- empleado\_ciudad\_residencia VARCHAR(100): Ciudad de residencia del empleado.
- empleado\_celular VARCHAR(20): Número de celular del empleado.
- empleado\_correo\_electronico VARCHAR(100): Correo electrónico del empleado.

### Actualización de Empleado por Cédula

**Nombre del Procedimiento:** actualizar\_empleado

**Descripción:** Actualiza la información de un empleado en la tabla empleado basado en su cédula.

#### Parámetros:

- p\_cedula VARCHAR(20): Cédula del empleado a actualizar.
- p\_sucursal\_id INT: ID de la sucursal del empleado.
- p\_nombre1 VARCHAR(100): Primer nombre del empleado.
- p\_nombre2 VARCHAR(100): Segundo nombre del empleado (puede ser NULL).
- p\_apellido1 VARCHAR(100): Primer apellido del empleado.

- p\_apellido2 VARCHAR(100): Segundo apellido del empleado (puede ser NULL).
- p\_direccion VARCHAR(200): Dirección del empleado.
- p\_ciudad\_residencia VARCHAR(100): Ciudad de residencia del empleado.
- p\_celular VARCHAR(20): Número de celular del empleado.
- p\_correo\_electronico VARCHAR(100): Correo electrónico del empleado.

### **Registro de Cliente**

**Nombre del Procedimiento:** registro\_cliente

**Descripción:** Inserta un nuevo registro de cliente en la tabla cliente.

**Parámetros:**

- p\_cedula VARCHAR(20): Cédula del cliente.
- p\_nombre1 VARCHAR(100): Primer nombre del cliente.
- p\_nombre2 VARCHAR(100): Segundo nombre del cliente (puede ser NULL).
- p\_apellido1 VARCHAR(100): Primer apellido del cliente.
- p\_apellido2 VARCHAR(100): Segundo apellido del cliente (puede ser NULL).
- p\_direccion VARCHAR(200): Dirección del cliente.
- p\_ciudad\_residencia VARCHAR(100): Ciudad de residencia del cliente.
- p\_celular VARCHAR(20): Número de celular del cliente.
- p\_correo\_electronico VARCHAR(100): Correo electrónico del cliente.

### **Actualización de Cliente por Cédula**

**Nombre del Procedimiento:** actualizar\_cliente

**Descripción:** Actualiza la información de un cliente en la tabla cliente basado en su cédula.

**Parámetros:**

- p\_cedula VARCHAR(20): Cédula del cliente a actualizar.
- p\_nombre1 VARCHAR(100): Primer nombre del cliente.
- p\_nombre2 VARCHAR(100): Segundo nombre del cliente (puede ser NULL).
- p\_apellido1 VARCHAR(100): Primer apellido del cliente.
- p\_apellido2 VARCHAR(100): Segundo apellido del cliente (puede ser NULL).
- p\_direccion VARCHAR(200): Dirección del cliente.
- p\_ciudad\_residencia VARCHAR(100): Ciudad de residencia del cliente.
- p\_celular VARCHAR(20): Número de celular del cliente.
- p\_correo\_electronico VARCHAR(100): Correo electrónico del cliente.

### **Registro de Vehículo**

**Nombre del Procedimiento:** registrar\_vehiculo

**Descripción:** Inserta un nuevo registro de vehículo en la tabla vehiculo, validando que el tipo de vehículo sea válido.

**Parámetros:**

- p\_tipo ENUM('sedán', 'compacto', 'camioneta platón', 'camioneta lujo', 'deportivo'): Tipo de vehículo.
- p\_placa VARCHAR(20): Placa del vehículo.
- p\_referencia VARCHAR(100): Referencia del vehículo.
- p\_modelo INT: Modelo del vehículo.
- p\_puertas INT: Número de puertas del vehículo.
- p\_capacidad INT: Capacidad del vehículo.
- p\_sunroof BOOLEAN: Indica si el vehículo tiene sunroof.
- p\_motor VARCHAR(50): Motor del vehículo.
- p\_color VARCHAR(30): Color del vehículo.

**Actualización de Vehículo por Placa**

**Nombre del Procedimiento:** actualizar\_vehiculo

**Descripción:** Actualiza la información de un vehículo en la tabla vehiculo basado en su placa, validando que el tipo de vehículo sea válido.

**Parámetros:**

- p\_placa VARCHAR(20): Placa del vehículo a actualizar.
- p\_tipo ENUM('sedán', 'compacto', 'camioneta platón', 'camioneta lujo', 'deportivo'): Tipo de vehículo.
- p\_referencia VARCHAR(100): Referencia del vehículo.
- p\_modelo INT: Modelo del vehículo.
- p\_puertas INT: Número de puertas del vehículo.
- p\_capacidad INT: Capacidad del vehículo.
- p\_sunroof BOOLEAN: Indica si el vehículo tiene sunroof.
- p\_motor VARCHAR(50): Motor del vehículo.
- p\_color VARCHAR(30): Color del vehículo.

**Registro de Sucursal**

**Nombre del Procedimiento:** registrar\_sucursal

**Descripción:** Inserta un nuevo registro de sucursal en la tabla sucursal.

**Parámetros:**

- p\_ciudad VARCHAR(100): Ciudad de la sucursal.
- p\_direccion VARCHAR(200): Dirección de la sucursal.
- p\_telefono\_fijo VARCHAR(20): Teléfono fijo de la sucursal.
- p\_celular VARCHAR(20): Celular de la sucursal.
- p\_correo\_electronico VARCHAR(100): Correo electrónico de la sucursal.

## **Actualización de Sucursal por ID**

**Nombre del Procedimiento:** actualizar\_sucursal

**Descripción:** Actualiza la información de una sucursal en la tabla sucursal basado en su ID, siempre que el ID de la sucursal exista.

**Parámetros:**

- p\_id INT: ID de la sucursal a actualizar.
- p\_ciudad VARCHAR(100): Ciudad de la sucursal.
- p\_direccion VARCHAR(200): Dirección de la sucursal.
- p\_telefono\_fijo VARCHAR(20): Teléfono fijo de la sucursal.
- p\_celular VARCHAR(20): Celular de la sucursal.
- p\_correo\_electronico VARCHAR(100): Correo electrónico de la sucursal.

## **Consulta de Disponibilidad de Vehículos**

**Nombre del Procedimiento:** consultar\_disponibilidad\_vehiculos

**Descripción:** Consulta la disponibilidad de vehículos para alquiler en función del tipo de vehículo, rango de precios y fechas de disponibilidad.

**Parámetros:**

- p\_tipo ENUM('sedán', 'compacto', 'camioneta platón', 'camioneta lujo', 'deportivo'): Tipo de vehículo a consultar.
- p\_precio\_min DECIMAL(10, 2): Precio mínimo del alquiler diario.
- p\_precio\_max DECIMAL(10, 2): Precio máximo del alquiler diario.
- p\_fecha\_inicio DATE: Fecha de inicio de disponibilidad.
- p\_fecha\_fin DATE: Fecha de fin de disponibilidad.

## **Registro de Alquiler de Vehículo**

**Nombre del Procedimiento:** registrar\_alquiler

**Descripción:** Registra el alquiler de un vehículo para un cliente específico en una sucursal, durante un período determinado.

**Parámetros:**

- p\_vehiculo\_id INT: ID del vehículo que se va a alquilar.
- p\_cliente\_id INT: ID del cliente que realiza el alquiler.
- p\_empleado\_id INT: ID del empleado que gestiona el alquiler.
- p\_sucursal\_salida\_id INT: ID de la sucursal desde la cual se alquila el vehículo.

- p\_sucursal\_llegada\_id INT: ID de la sucursal a la cual se debe devolver el vehículo.
- p\_fecha\_salida DATE: Fecha de inicio del alquiler.
- p\_fecha\_llegada DATE: Fecha de fin del alquiler.
- p\_fecha\_esperada\_llegada DATE: Fecha esperada de devolución del vehículo.
- p\_valor\_alquiler\_semana DECIMAL(10, 2): Valor del alquiler por semana.
- p\_valor\_alquiler\_dia DECIMAL(10, 2): Valor del alquiler por día.
- p\_porcentaje\_descuento DECIMAL(5, 2): Porcentaje de descuento aplicado al alquiler (opcional).
- p\_valor\_cotizado DECIMAL(10, 2): Valor cotizado inicialmente para el alquiler.
- p\_valor\_pagado DECIMAL(10, 2): Valor pagado por el alquiler hasta el momento del registro.

### **Consulta de Historial de Alquileres por Cliente**

**Nombre del Procedimiento:** consultar\_historial\_alquileres\_cliente

**Descripción:** Consulta el historial de alquileres de un cliente específico basado en su cédula.

**Parámetros:**

- p\_cedula\_cliente VARCHAR(20): Cédula del cliente para consultar el historial.

**Permisos:**

A cada usuario se le otorgan unos permisos los cuales les dejara utilizar funciones específicas dependiendo de qué tipo de usuario sea, ya que va a contar con 2 usuarios principales que serían:

- **Empleado:** El Empleado va a tener acceso a registrar y actualizar lo que son, empleados, sucursales y vehículos.
- **Cliente:** El cliente tendrá acceso a registrarse, consultar disponibilidad de vehículos, alquilar vehículo y por último el historial de sus alquileres que este lo hará a través de su id.

**Triggers y Event:**

Se crearon dos tablas adicionales los cuales la primera tabla que va relacionada con 2 triggers nos permite visualizar la hora en la que se hizo un registro o una actualización de un empleado y si se hizo una actualización o un registro.



```
DELIMITER //
```

```
CREATE TRIGGER after_empleado_insert
```

```
AFTER INSERT ON empleado
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO inserciones_actualizaciones_empleados (id, nombrel,
```

```
nombre2, apellido1, apellido2, direccion, ciudad_residencia, celular,
```

```
correo_electronico, action)
```

```
    VALUES (NEW.id, NEW.nombrel, NEW.nombre2, NEW.apellido1,
```

```
NEW.apellido2, NEW.direccion, NEW.ciudad_residencia, NEW.celular,
```

```
NEW.correo_electronico, 'INSERT');
```

```
END //
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE TRIGGER after_empleado_update
```

```
AFTER UPDATE ON empleado
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    INSERT INTO inserciones_actualizaciones_empleados (id, nombrel,
```

```
nombre2, apellido1, apellido2, direccion, ciudad_residencia, celular,
```

```
correo_electronico, action)
```

```
    VALUES (NEW.id, NEW.nombrel, NEW.nombre2, NEW.apellido1,
```

```
NEW.apellido2, NEW.direccion, NEW.ciudad_residencia, NEW.celular,
```

```
NEW.correo_electronico, 'UPDATE');
```

```
END //
```

```
DELIMITER ;
```

La segunda tabla es un back up de empleados para que se guarde la información en una tabla por un día contando desde el momento que se actualizo a el nuevo empleado con el objetivo de que si se comete algún error tengan tiempo de corregirlo.

```
DELIMITER //
```

```
CREATE EVENT IF NOT EXISTS daily_employee_backup
```

```
ON SCHEDULE EVERY 1 DAY
```

```
STARTS CURRENT_TIMESTAMP
```

```
DO
```

```
BEGIN
```

```
    INSERT INTO backup_empleados (nombre1, nombre2, apellido1,
```

```
apellido2, direccion, ciudad_residencia, celular, correo_electronico)
```

```
    SELECT nombre1, nombre2, apellido1, apellido2, direccion,
```

```
ciudad_residencia, celular, correo_electronico
```

```
    FROM empleado;
```

```
END //
```

```
DELIMITER ;
```