**Auto\_Rental**

**Valentina Isabel Molina Lopera**

**T2**

**Pedro Felipe Gómez Bonilla**

**Campuslands**

**Ruta Java**

**Tibú, Norte de Santander**

**2024**

1

Tabla de contenido

Introducción.....................................................................................................................................

Caso de Estudio...............................................................................................................................

Instalación General..........................................................................................................................

Planificación....................................................................................................................................

Ejecución.......................................................................................................................................

Construcción del Modelo Conceptual...................................................................................

Descripción..........................................................................................................................

Las Entidades y Atributos...............................................................................................

Relaciones y Cardinalidades..........................................................................................

Gráfica.................................................................................................................................

Construcción del Modelo Lógico.........................................................................................

Descripción..........................................................................................................................

Las Entidades y Atributos.............................................................................................

Relaciones y Cardinalidades........................................................................................

Gráfica.................................................................................................................................

Normalización del Modelo Lógico..............................................................................................

Primera Forma Normal (1FN) ..............................................................................................

Descripción..........................................................................................................................

Descripción Técnica............................................................................................................

Gráfica.................................................................................................................................

Segunda Forma Normal (2FN) ...........................................................................................

Descripción..........................................................................................................................

Descripción Técnica............................................................................................................

Gráfica.................................................................................................................................

Tercera Forma Normal (3FN) .............................................................................................

Descripción..........................................................................................................................

Descripción Técnica............................................................................................................

Gráfica.................................................................................................................................

Construcción del Modelo Físico..........................................................................................

Descripción.........................................................................................................................

Tablas.................................................................................................................................

Construcción del Diagrama UML.......................................................................................

Descripción........................................................................................................................

Gráfica................................................................................................................................

Inserciones de Datos..........................................................................................................

Descripción.........................................................................................................................

Consultas de Datos............................................................................................................

Descripción........................................................................................................................

# Introducción

El objetivo principal de este proyecto es el diseño y desarrollo de una base de datos para el sistema de información de la empresa AutoRental. Esta base de datos permitirá gestionar de toda la información relacionada con la flota de vehículos, los clientes y los procesos de alquiler.

Esta empresa actualmente tiene 5 sucursales estratégicamente ubicadas en distintas ciudades del país.

Uno de los puntos claves del servicio que ofrece AutoRental es la flexibilidad de alquilar un vehículo en una sucursal y devolverlo en otra, para lo que se necesitara una gestión logística más compleja.

Cuenta con una flota propia de vehículos de diferentes tipos, modelos

(año), capacidad, etc.

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 enTtrega contratada, se cobrarán los días

adicionales con un incremento del 8%.

# Caso de Estudio

La empresa AutoRental nos ha solicitado crear una base de datos para gestionar de manera eficiente la información de los alquileres.

Estructuración de requerimientos:

1. Sucursales: ciudad y dirección donde se ubica, teléfono fijo, celular y correo electrónico.
2. Empleados: sucursal donde labora, cédula, nombres, apellidos, dirección y ciudad de

residencia, celular y correo electrónico.

1. Clientes: cédula, nombres, apellidos, dirección y ciudad de residencia, celular y correo

electrónico.

1. Vehículos: tipo de vehículo, placa, referencia, modelo, puertas, capacidad, sunroof, motor, color.
2. 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 pordía, porcentaje de descuento, valor cotizado y valor pagado.

Con la información dada anteriormente se realizará la base de datos.

# Planificación

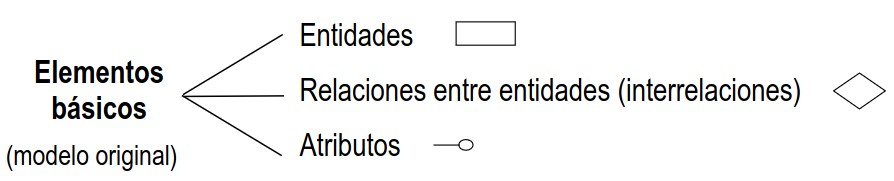
## Ejecución

Despues de haber analizado la información requerida por la empresa de AutoRental, se inicia la creación del modelo conceptual. Este modelo no sarav una descripción detallada de las necesidades que están detrás del diseño de la base de datos.

### Construcción del Modelo Conceptual

Hemos diseñado el modelo conceptual identificando cada una de las entidades, sus atributos y las relaciones entre ellas. Este modelo nos da una visión clara y estructurada de cómo se organizarán y conectarán los diferentes elementos de la base de datos.

Para entender el diseño del modelo conceptual, se debe tener en cuenta los elementos básicos de un modelo original.



### Descripción

#### **Las Entidades y Atributos**

1. **empleado:**

* idempleado: identificador único de empleado.
* nombre: nombre del empleado.
* apellido: apellido del empleado.
* dirección: dirección de residencia.
* cuidad: cuidad de residencia.
* celular: numero celular del empleado.
* email: correo electrónico del empleado.
* cedula: numero de la cedula de ciudadanía del empleado.
* Sucursal\_idsucursal: Objeto traído de otra tabla para la conexión de las mismas

1. **cliente:**
   * + idcliente: identificador único del cliente.
     + cedula: numero de la cedula de ciudadanía del empleado.
     + nombre: nombre del cliente.
     + apellido: apellido del cliente.
     + dirección: dirección de residencia del cliente.
     + cuidad: cuidad de residencia del cliente.
     + email: correo electrónico del cliente.
2. **vehiculo:**
   * + idvehiculo: identificados único del vehiculo.
     + tipo\_vehiculo: (sedán, hatchback, deportivo.etc…)
     + placa: placa del vehiculo.
     + Referencia: año del vehiculo.
     + modelo: modelo el vehiculo.
     + puertas: Número de puertas del vehículo.
     + sunroof: Indica si el vehículo tiene techo solar (sí/no).
     + motor: Nombre o código del motor.
     + color: color principal del vehiculo.
     + capacidad: Capacidad de pasajeros o carga, según el tipo de vehículo.
3. **sucursal:**
   * + idsucursal: identificador único de la sucursal.
     + cuidad: ciudad donde queda la sucursal.
     + dirección: dirección de la sucursal.
     + teléfono\_fijo: teléfono fijo de la sucursal.
     + celular: numero celular de la sucursal.
     + email: correo electrónico de la sucursal.
4. **alquiler:**
   * + idalquiler: identificador único del alquiler
     + fecha\_salida: fecha en la que el vehiculo inicia el alquiler.
     + fecha\_esperada: fecha a la que se espera que el cliente haga la entrega del vehiculo.
     + fecha\_entrega: fecha en la que el cliente hace la entrega del vehiculo.
     + sucursal\_salida: sucursal de donde es alquilado y sale el vehiculo.
     + sucursal\_entrega: sucursal en donde el cliente hace entrega del vehiculo.
     + valor\_dia: precio que se le cobrara por dia al cliente.
     + valor\_semanal: monto que se le cobrara por semana al cliente.
     + valor\_cotizado: monto cotizado por alquiler
     + valor\_pagado: monto que el cliente pago
     + descuento: monto que se descontara del alquiler
     + cliente\_idcliente: objeto traído de otra tabla para la conexión de las mismas
     + vehiculo\_idvehiculo: objeto traído de otra tabla para la conexión de las mismas
     + sucursal\_idsucursal: objeto traído de otra tabla para la conexión de las mismas
     + empleado\_idempleado: objeto traído de otra tabla para la conexión de las mismas

#### **Relaciones y Cardinalidades**

Se realizó las relaciones y cardinalidades respectivas del modelo conceptual con sus entidades para tener mejor visualización de la base de datos:

1. **sucursal - empleados:**

❖ **Relación**: Una sucursal "**Tiene**", muchos empleados y muchos empleados “**Tienen**” una misma sucursal.

* **Cardinalidad**: 1-N (uno a muchos).

1. **sucursal - alquiler:**

❖ **Relación**: Una sucursal "**tiene**", muchos alquileres y muchos alquileres “**Tienen**” una misma sucursal.

* **Cardinalidad**: 1-N (uno a muchos).

1. **empleado - alquiler:**

❖ **Relación**: Un empleado puede "**realizar**", muchos alquileres y muchos alquileres son “**realizados**” en misma sucursal.

* **Cardinalidad**: 1-N (uno a muchos).

1. **cliente - alquiler:**

❖ **Relación**: Un cliente puede "**realizar**", muchos alquileres y muchos alquileres son “**realizados**” con un mismo cliente.

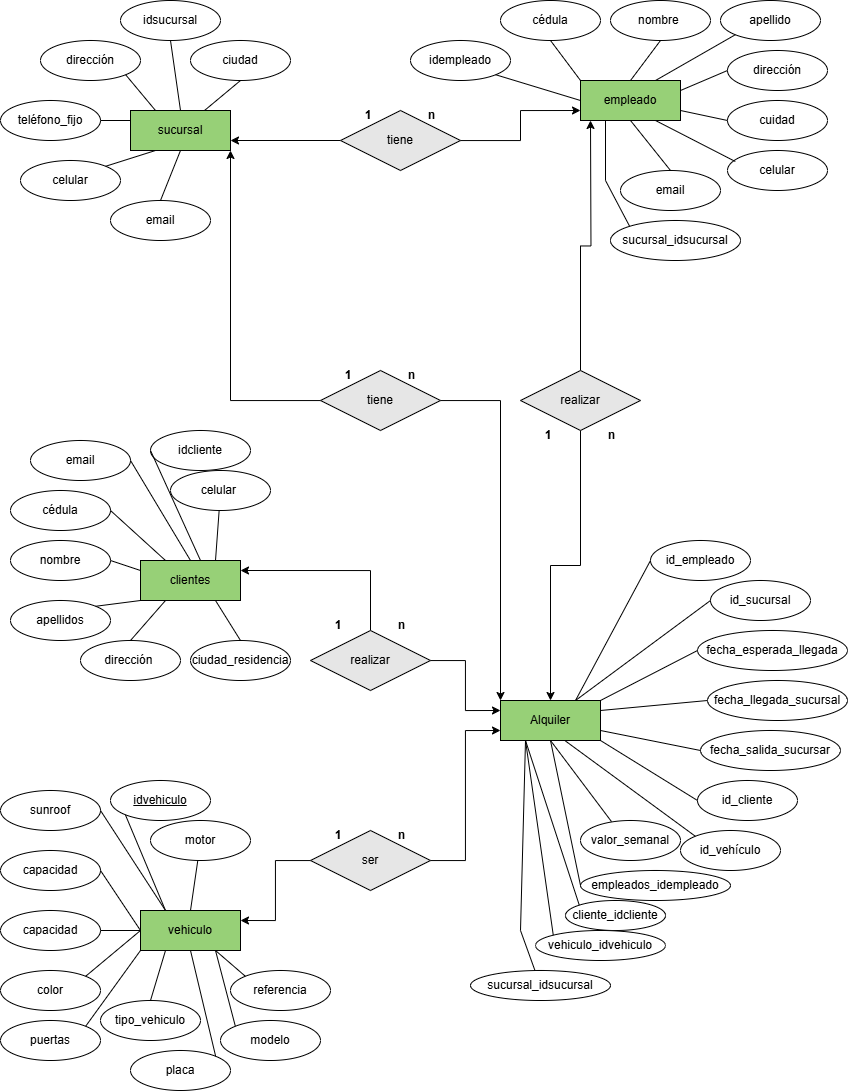
* **Cardinalidad**: 1-N (uno a muchos).

1. **vehiculo - alquiler:**

❖ **Relación**: un vehiculo puede "ser", alquilado muchas veces y muchos alquileres pueden “**ser**” realizados con un vehiculo.

* **Cardinalidad**: 1-N (uno a muchos).

Gráfica



### Construcción del Modelo Lógico

Se ha diseñado el modelo lógico teniendo en cuenta el modelo conceptual, incorporando detalles más específicos como las características de cada atributo, incluidas las claves primarias, foráneas y las relaciones de cardinalidad.

### Descripción

#### **Las Entidades y Atributos**

1. **empleados**

* idempleado: int - auto\_increment – primary key.
* nombre: varchar (45) - not null
* apellido: varchar (45) - not null
* dirección: varchar (45) - not null
* cuidad: varchar (45) - not null
* celular: varchar – not null
* email: varchar (45) - not null
* cedula: varchar – not null
* sucursal\_idsucursal: int - not null foreign key

1. **cliente:**
   * + idcliente: int - auto\_increment – primary key
     + cedula: varchar – not null
     + nombre: varchar (30) – not null
     + apellido: varchar (50) – not null
     + dirección: varchar (30) – not null
     + cuidad: varchar (50) - not null
     + email: varchar (60) – not null
     + celular: varchar (45) - not null
2. **vehiculo:**
   * + idvehiculo: int - auto\_increment – primary key
     + tipo\_vehiculo: varchar (45) - not null
     + placa: varchar (45) - not null
     + referencia: varchar (45) - not null
     + modelo: varchar (45) - not null
     + puertas: varchar (45) - not null
     + sunroof: varchar (45) - not null
     + motor: varchar (45) - not null
     + color: varchar (45) - not null
     + capacidad: varchar (45) - not null
3. **sucursal:**
   * + idsucursal: int - auto\_increment – primary key
     + cuidad: varchar (45) - not null
     + dirección: varchar (45) - not null
     + teléfono\_fijo: varchar (45) – not null
     + celular: varchar (45) - not null
     + email: varchar (60) - not null
4. **alquiler:**
   * + idalquiler: int - auto\_increment – primary key
     + fecha\_salida: date - not null
     + fecha\_esperada: date - not null
     + fecha\_entrega: date - not null
     + sucursal\_salida: varchar (45) - not null
     + sucursal\_entrega: varchar (45) - not null
     + valor\_dia: decimal – not null
     + valor\_semanal: decimal – not null
     + valor\_cotizado: decimal – not null
     + valor\_pagado: decimal – not null
     + descuento: decimal
     + cliente\_idcliente: int – not null - not null foreign key
     + vehiculo\_idvehiculo: int – not null- not null foreign key
     + sucursal\_idsucursal: int – not null- not null foreign key
     + empleado\_idempleado: int – not null - not null foreign key

#### **Relaciones y Cardinalidades**

Se realizó las relaciones y cardinalidades respectivas del modelo lógico con sus entidades para tener mejor visualización de la base de datos:

1. **sucursal - empleados:**

❖ Una sucursal "**Tiene**", muchos empleados y muchos empleados “**Tienen**” una misma sucursal.

* 1-N (uno a muchos).



1. **sucursal - alquiler:**

❖ Una sucursal "**tiene**", muchos alquileres y muchos alquileres “**Tienen**” una misma sucursal.

* 1-N (uno a muchos).



1. **empleados - alquiler:**

❖ Un empleado puede "**realizar**", muchos alquileres y muchos alquileres son “**realizados**” en misma sucursal.

* 1-N (uno a muchos).



1. **cliente - alquiler:**

❖ Un cliente puede "**realizar**", muchos alquileres y muchos alquileres son “**realizados**” con un mismo cliente.

* 1-N (uno a muchos).



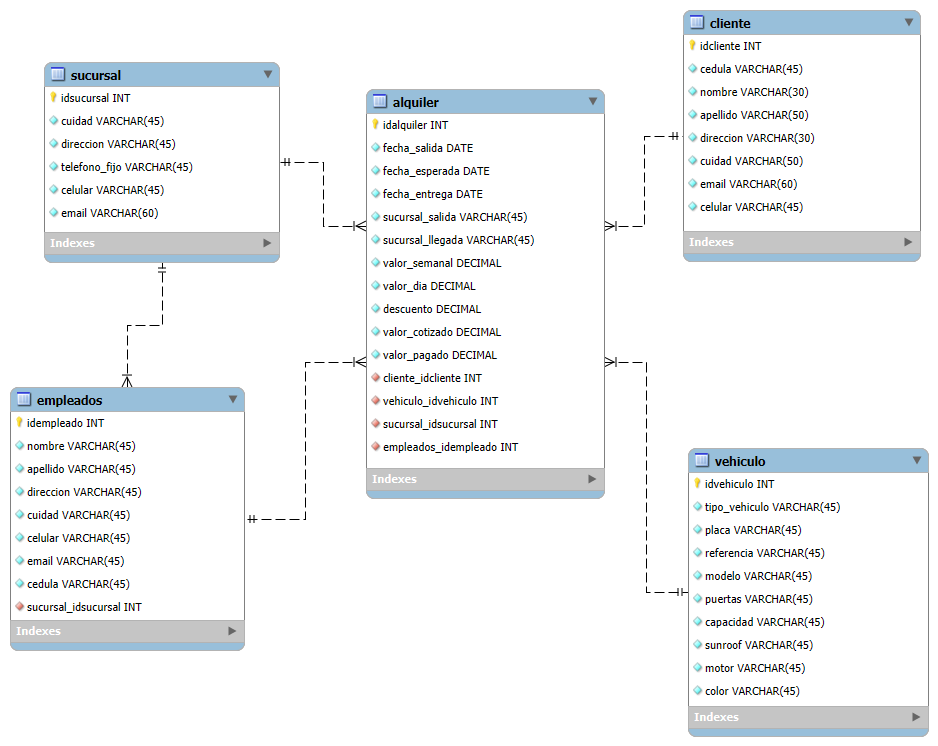
1. **vehiculo - alquiler:**

❖ Un vehiculo puede "ser", alquilado muchas veces y muchos alquileres pueden “**ser**” realizados con un vehiculo.

* 1-N (uno a muchos).



### Gráfica



## Normalización del Modelo Lógico

Realizaremos la normalización de las tablas anteriores para dar una vista mas organizada, presiza y eficiente de los datos.

### Primera Forma Normal (1FN)

Una tabla está en 1FN si cumple con los siguientes criterios:

* Todos los atributos contienen valores atómicos (indivisibles).
* No debe haber grupos repetitivos de columnas.
* Cada columna debe contener un solo valor en cada fila.

### Descripción

La primera forma normal, es el primer nivel de normalización en el diseño de la base de datos que se aplicará a las tablas de la base de datos para garantizar la organización de los datos de manera que evite redundancias y asegure la consistencia de la información.

### Descripción Técnica

1. **empleados:**

❖ Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.



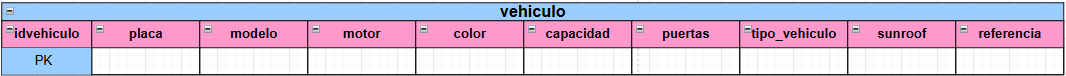
1. **cliente:**

❖ Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.



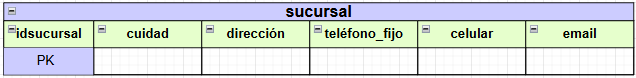
1. **vehiculo:**

❖ Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.



1. **sucursal:**

❖ Se encuentra en 1FN, ya que cuenta con las claves primarias compuestas y cada columna tiene valores únicos y no son repetitivos.



1. **alquiler:**
   * Se encuentra en 1FN, ya que cuenta con una clave primaria única y cada columna tiene valores únicos y no son repetitivos.



### Segunda Forma Normal (2FN)

Una tabla está en 2FN si cumple con los siguientes criterios:

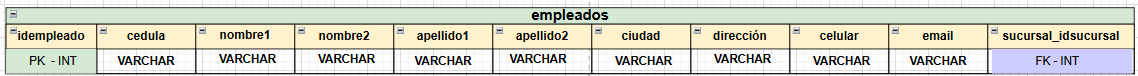
* Está en 1FN.
* Todos los atributos que no son claves (no pertenecientes a una clave primaria compuesta) dependen completamente de la clave primaria.

### Descripción

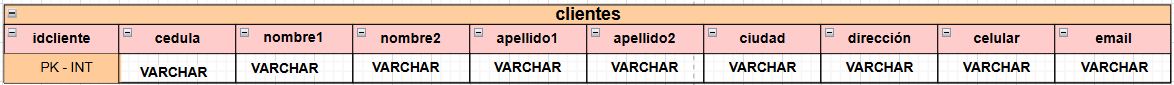
La segunda forma normal, es el segundo nivel de normalización en el diseño de la base de datos que se aplicará a las tablas de una base de datos que ya cumplen con la primera forma normal y lleva a cabo la eliminación de dependencias parciales dentro de una tabla.

### Descripción Técnica

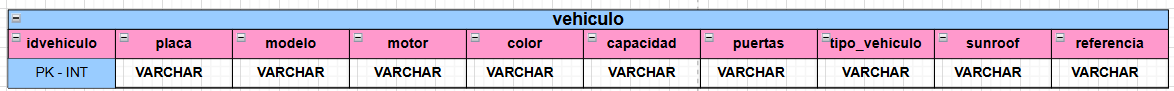
1. **entidad:**
   * Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.



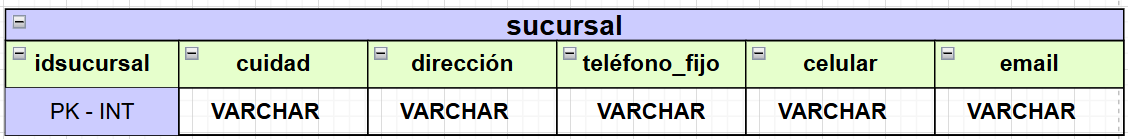
1. **cliente:**
   * Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.



1. **vehiculo:**
   * Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.



1. **sucursal:**
   * Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.



1. **alquiler:**
   * Se encuentra en 2FN, ya que cuenta con una clave primaria única y cada columna depende completamente de esa clave primaria.

### 

### Tercera Forma Normal (3FN)

Una tabla está en 3NF si cumple con los siguientes criterios:

* Está en 2NF.
* No hay dependencias transitivas: ningún atributo no clave depende de otro atributo no clave.

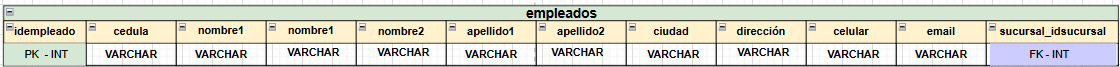
### Descripción

La tercera forma normal, es el tercer nivel de normalización en el diseño de la base de datos que se aplicará a las tablas de una base de datos que ya cumplen con la segunda forma normal y se enfoca en la eliminación de dependencias transitivas, evitando que un atributo no clave dependa de otro no clave.

### Descripción Técnica

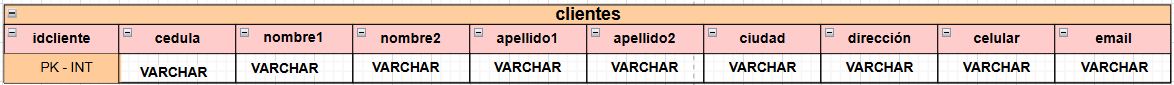
1. **entidad:**

❖ Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

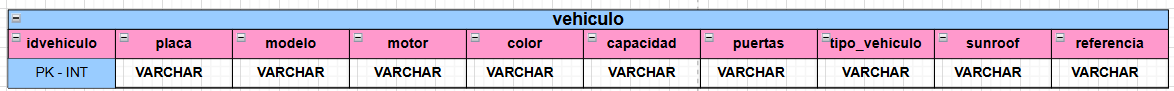


1. **departamento:**

❖ Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

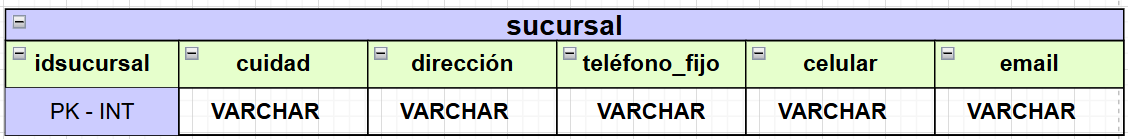


1. **parque:**
   * Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.

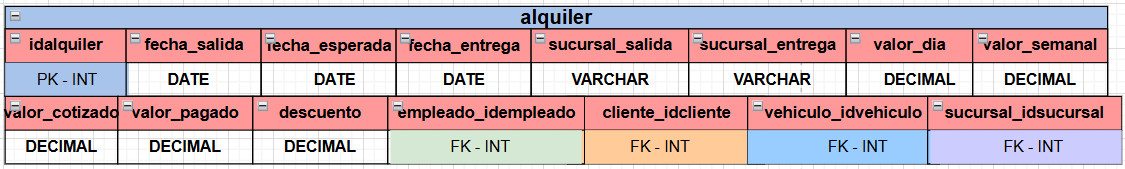


1. **parque\_departamento:**

Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.



1. **area:**
   * Se encuentra en 3FN, ya que está en la 2FN y en cada columna no hay dependencias transitivas con la clave primaria.



### Construcción del Modelo Físico

Se diseñó el modelo físico considerando el modelo lógico que incluye todas las entidades, sus atributos y las relaciones entre ellas. Además, este modelo incorpora los tipos de datos de los atributos previamente definidos, los cuales fueron estructurados en tablas utilizando el lenguaje de un Sistema de Gestión de Bases de Datos (SGBD) compatible con la plataforma MySQL.

### Descripción

El modelo físico se diseñó para funcionar en MySQL, donde cada entidad se representa como una tabla compuesta por sus atributos correspondientes, organizados en columnas con tipos de datos específicos según sea necesario.

### Tablas

Para crear la base de datos utilice el siguiente comando:

##### create database AutoRental;

Para utilizar la base de datos ocupe el siguiente comando:

##### use AutoRental;

Comenzaremos creando las tablas junto con sus tipos de datos correspondientes. Para esto, utiliza los siguientes comandos:

1. Creación de la tabla **sucursal**

|  |
| --- |
| create table sucursal ( |
| idsucursal INT AUTO\_INCREMENT PRIMARY KEY, |
| ciudad VARCHAR (45) NOT NULL,  direccion VARCHAR (45) NOT NULL,  telefono\_fijo VARCHAR (45) NOT NULL,  cellular VARCHAR (45) NOT NULL,  email VARCHAR (45) NOT NULL  ); |

1. Creación de la tabla **empleado**

|  |
| --- |
| create table empleado ( |
| idempleado INT AUTO\_INCREMENT PRIMARY KEY, |
| cedula VARCHAR(45) NOT NULL, |
| nombre1 VARCHAR (45) NOT NULL, |
| nombre2 VARCHAR (45) NOT NULL,  apellido1 VARCHAR (45) NOT NULL,  apellido2 VARCHAR (45) NOT NULL,  ciudad VARCHAR (45) NOT NULL, |
| direccion VARCHAR (45) NOT NULL, |
| email VARCHAR (45) NOT NULL,  sucursal\_idsucursal INT NOT NULL, |
| Foreign Key (sucursal\_idsucursal) REFERENCES sucursal(idsucursal) |
| ); |

1. Creación de la tabla **cliente**

|  |
| --- |
| create table cliente ( |
| idcliente INT AUTO\_INCREMENT PRIMARY KEY,  cedula VARCHAR(45) NOT NULL, |
| nombre1 VARCHAR(45) NOT NULL, |
| nombre2 VARCHAR (45) NOT NULL,  apellido1 VARCHAR (45) NOT NULL,  apellido2 VARCHAR (45) NOT NULL,  cuidad VARCHAR (45) NOT NULL,  dirección VARCHAR (45) NOT NULL,  celular VARCHAR (45) NOT NULL,  email VARCHAR (45) NOT NULL  ); |

1. Creación de la tabla **vehiculo**

|  |
| --- |
| create table vehiculo ( |
| idvehiculo INT AUTO\_INCREMENT PRIMARY KEY,  placa VARCHAR(45) NOT NULL, |
| modelo VARCHAR(45) NOT NULL, |
| motor VARCHAR (45) NOT NULL,  color VARCHAR (45) NOT NULL,  capacidad VARCHAR (45) NOT NULL,  puertas VARCHAR (45) NOT NULL,  tipo\_vehiculo VARCHAR (45) NOT NULL,  suroof VARCHAR (45) NOT NULL,  referencia VARCHAR (45) NOT NULL  ); |

1. Creación de la tabla a **alquiler**

|  |
| --- |
| create table alquiler ( |
| idalquiler INT AUTO\_INCREMENT PRIMARY KEY, |
| Fecha\_salida DATE NOT NULL, |
| Fecha\_esperada DATE NOT NULL,  Fecha\_entrega DATE NOT NULL, |
| Sucursal\_salida VARCHAR (45) NOT NULL,  Sucursal\_entrega VARCHAR (45) NOT NULL,  valor\_dia DECIMAL NOT NULL,  valor\_semanal DECIMAL NOT NULL, |
| valor\_cotizado DECIMAL NOT NULL, |
| descuento DECIMAL NOT NULL,  sucursal\_idsucursal INT NOT NULL, |
| Foreign Key (sucursal\_idsucursal) REFERENCES sucursal(idsucursal)  Cliente\_idcliente INT NOT NULL, |
| Foreign Key (Cliente\_idcliente) REFERENCES Cliente (idcliente)  Vehiculo\_idvehiculo INT NOT NULL,  Foreign Key (Vehiculo\_idvehiculo) REFERENCES Vehiculo (idvehiculo)  Empleado\_idempleado INT NOT NULL,  Foreign Key (Empleado\_idempleado) REFERENCES Empleado(idempleado)  ); |

Finalmente, para listar las tablas anteriormente creadas utilizamos este comando:

SHOW TABLES;

### Construcción del Diagrama UML

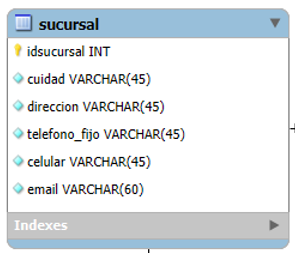
### Para comprender mejor los diseños, la arquitectura del código y la implementación propuesta, se ha creado un diagrama uml estandarizado como referencia. Al utilizar este enfoque, podemos obtener una comprensión detallada de cómo se administrarán las consultas, las funciones y los usuarios de la base de datos. Al hacer esto, podemos garantizar que la implementación sea eficiente y se alinee con los requisitos del sistema.

### Descripción

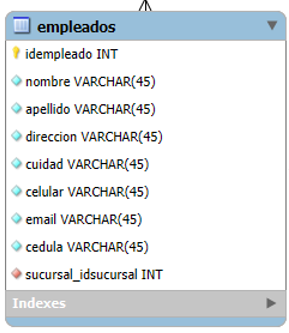
El diagrama uml fue creado para exhibir la estructura de cada tabla y sus relaciones con gran detalle. El tipo de datos asociado con cada atributo se muestra claramente en este diagrama, junto con la identificación de las claves primaria y externa. Además, describe las características esenciales de los elementos, ofreciendo una visión precisa y completa de la base de datos. Este nivel de detalle permite la comprensión e implementación técnicas, asegurando que todos los componentes y sus interrelaciones estén correctamente definidos y alineados con los requisitos del sistema.

Comenzaremos creando las tablas junto con sus tipos de datos correspondientes:

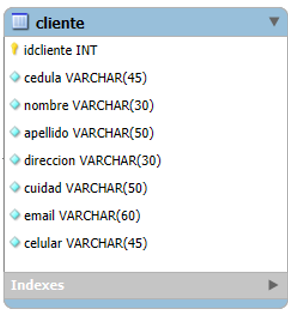
1. Tabla: sucursal



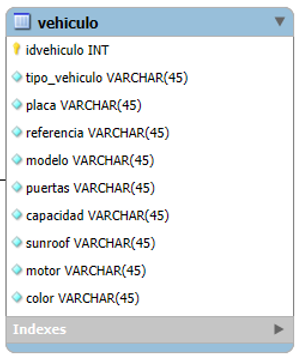
1. Tabla: empleados



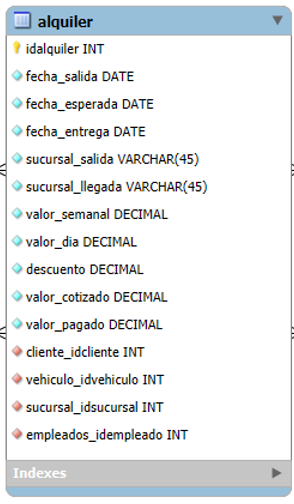
1. Tabla: cliente



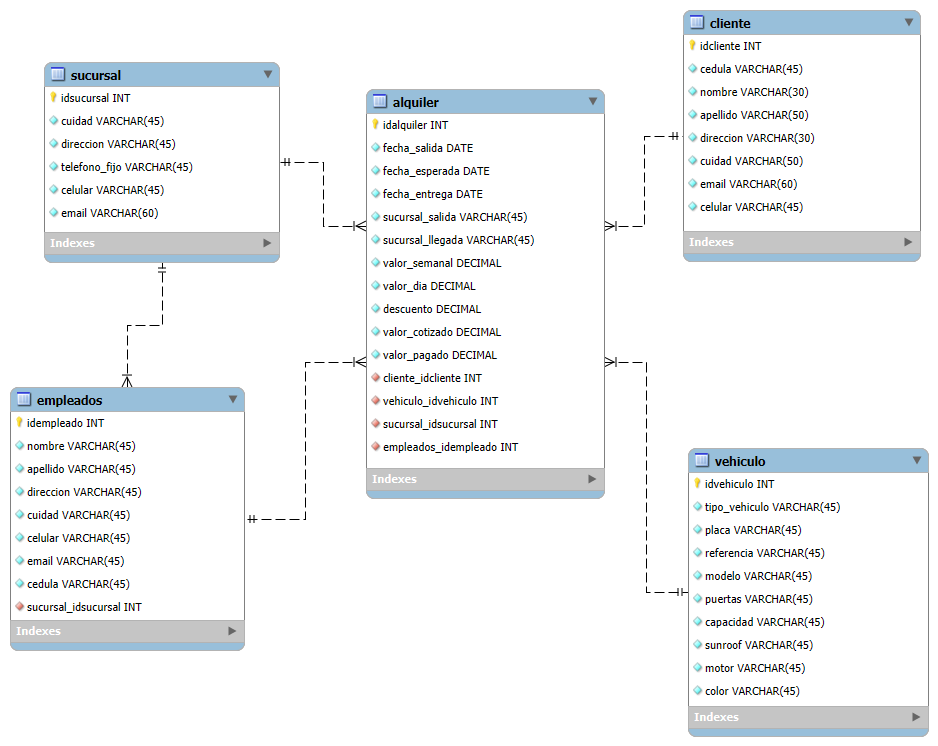
1. Tabla: vehiculo



1. Table: alquiler



### Gráfica



### Inserciones de Datos

La inserción de datos en las tablas es una parte fundamental para la prueba de operatividad. Para insertar datos en una tabla específica, se utiliza la siguiente sintaxis:

|  |
| --- |
| INSERT INTO nombre\_de\_tabla (columna1, columna2, columna3, ...) |
| VALUES (valor1, valor2, valor3, ...); |

* **nombre\_de\_tabla**: El nombre de la tabla donde se insertarán los datos.
* **columna1, columna2, columna3:** Son los atributos de las tablas en los cuales se ingresarán los datos específicamente.
* **VALUES (valor1, valor2, valor3)**: Se ingresan los valores que se desea en cada columna específica, los valores están en el mismo orden que las columnas.

### Descripción

**Sucursal**: Para ingresar datos en la tabla ‘sucursal’, la cual muestra (idsucursal, ciudad, direccion, telefono\_fijo, cellular, email) se deberá emplementar el siguiente comando:

|  |
| --- |
| INSERT INTO sucursal (idsucursal, ciudad, direccion, telefono\_fijo, cellular, email) VALUES  (1'Bogotá', 'Calle 123', '6012345678', '3001234567', 'bogota@sucursal.com'), (2'Medellín', 'Avenida 456', '6045678901', '3012345678', 'medellin@sucursal.com'), (3'Cali', 'Carrera 789', '6023456789', '3023456789', 'cali@sucursal.com'), (4'Barranquilla', 'Diagonal 101', '6056789012', '3034567890', 'barranquilla@sucursal.com'),  (5'Cartagena', 'Transversal 202', '6057890123', '3045678901', 'cartagena@sucursal.com'  ); |

**Empleado**: Para ingresar datos en la tabla ‘empleado, la cual muestra (cedula, nombre1, nombre2, apellido1, apellido2, ciudad, direccion, email, sucursal\_idsucursal) se emplea el siguiente comando:

|  |
| --- |
| -- Empleados para la sucursal 1 (Bogotá)  INSERT INTO empleado (idsucursal, cedula, nombre1, nombre2, apellido1, apellido2, ciudad, direccion, email, sucursal\_idsucursal) VALUES  (1'123456789', 'Juan', 'Carlos', 'Pérez', 'López', 'Bogotá', 'Calle 1', 'juan.carlos@example.com', 1),  (2'987654321', 'María', 'Fernanda', 'Rodríguez', 'García', 'Bogotá', 'Calle 2', 'maria.fernanda@example.com', 1),  (3'456789123', 'Luis', 'Alberto', 'Martínez', 'Díaz', 'Bogotá', 'Calle 3', 'luis.alberto@example.com', 1),  (4'654321987', 'Ana', 'Patricia', 'Gómez', 'Pérez', 'Bogotá', 'Calle 4', 'ana.patricia@example.com', 1),  (5'789123456', 'Carlos', 'Andrés', 'Hernández', 'Ortiz', 'Bogotá', 'Calle 5', 'carlos.andres@example.com', 1);  -- Empleados para la sucursal 2 (Medellín)  INSERT INTO empleado (cedula, nombre1, nombre2, apellido1, apellido2, ciudad, direccion, email, sucursal\_idsucursal) VALUES  (6'112233445', 'Sofía', 'Andrea', 'López', 'Jiménez', 'Medellín', 'Calle 6', 'sofia.andrea@example.com', 2),  (7'998877665', 'Miguel', 'Ángel', 'Sánchez', 'Ramírez', 'Medellín', 'Calle 7', 'miguel.angel@example.com', 2),  (8'556677889', 'Lucía', 'María', 'Ramírez', 'Torres', 'Medellín', 'Calle 8', 'lucia.maria@example.com', 2),  (9'223344556', 'José', 'Luis', 'Torres', 'Vargas', 'Medellín', 'Calle 9', 'jose.luis@example.com', 2),  (10'665544332', 'Elena', 'Isabel', 'Flores', 'Gómez', 'Medellín', 'Calle 10', 'elena.isabel@example.com', 2);  -- Empleados para la sucursal 3 (Cali)  INSERT INTO empleado (cedula, nombre1, nombre2, apellido1, apellido2, ciudad, direccion, email, sucursal\_idsucursal) VALUES  (11'321654987', 'Andrés', 'Felipe', 'Moreno', 'López', 'Cali', 'Calle 11', 'andres.felipe@example.com', 3),  (12'741852963', 'Paula', 'Andrea', 'Rojas', 'Martínez', 'Cali', 'Calle 12', 'paula.andrea@example.com', 3),  (13'852963741', 'Fernando', 'José', 'Castro', 'Hernández', 'Cali', 'Calle 13', 'fernando.jose@example.com', 3),  (14'963741852', 'Gloria', 'Elena', 'Vargas', 'Díaz', 'Cali', 'Calle 14', 'gloria.elena@example.com', 3),  (15'741963852', 'Martín', 'Antonio', 'Méndez', 'Pérez', 'Cali', 'Calle 15', 'martin.antonio@example.com', 3);  -- Empleados para la sucursal 4 (Barranquilla)  INSERT INTO empleado (cedula, nombre1, nombre2, apellido1, apellido2, ciudad, direccion, email, sucursal\_idsucursal) VALUES  (16'258147369', 'Laura', 'Marcela', 'Cruz', 'Gómez', 'Barranquilla', 'Calle 16', 'laura.marcela@example.com', 4),  (17'369258147', 'Pedro', 'Juan', 'Ortiz', 'Ramírez', 'Barranquilla', 'Calle 17', 'pedro.juan@example.com', 4),  (18'147369258', 'Natalia', 'Fernanda', 'Guerrero', 'López', 'Barranquilla', 'Calle 18', 'natalia.fernanda@example.com', 4),  (19'753951846', 'Felipe', 'Andrés', 'Muñoz', 'Martínez', 'Barranquilla', 'Calle 19', 'felipe.andres@example.com', 4),  (20'951846753', 'Camila', 'Isabel', 'Suárez', 'Hernández', 'Barranquilla', 'Calle 20', 'camila.isabel@example.com', 4);  -- Empleados para la sucursal 5 (Cartagena)  INSERT INTO empleado (cedula, nombre1, nombre2, apellido1, apellido2, ciudad, direccion, email, sucursal\_idsucursal) VALUES  (21'159357486', 'Jorge', 'Luis', 'Jiménez', 'Torres', 'Cartagena', 'Calle 21', 'jorge.luis@example.com', 5),  (22'357159486', 'Patricia', 'María', 'Fernández', 'Vargas', 'Cartagena', 'Calle 22', 'patricia.maria@example.com', 5),  (23'486357159', 'Ricardo', 'Alberto', 'Ruiz', 'Gómez', 'Cartagena', 'Calle 23', 'ricardo.alberto@example.com', 5),  (24'597486321', 'Adriana', 'Marcela', 'Delgado', 'Martínez', 'Cartagena', 'Calle 24', 'adriana.marcela@example.com', 5),  (25'123987654', 'David', 'Alejandro', 'García', 'Pérez', 'Cartagena', 'Calle 25', 'david.alejandro@example.com', 5); |

**Cliente**: Para ingresar datos en la tabla ‘cliente’, la cual muestra idcliente, cedula, nombre1, nombre2, apellido1, apellido2, cuidad, dirección, celular, email), se emplea el siguiente comando:

|  |
| --- |
| INSERT INTO cliente (idcliente, cedula, nombre1, nombre2, apellido1, apellido2, cuidad, dirección, celular, email) VALUES |
| (1,'1234567890', 'Juan', 'Carlos', 'Pérez', 'López', 'Bogotá', 'Calle 1', '3001234567', 'juan.carlos@example.com'), |
| (2,'2345678901', 'María', 'Fernanda', 'Rodríguez', 'García', 'Medellín', 'Calle 2', '3012345678', 'maria.fernanda@example.com'),  (3,'3456789012', 'Luis', 'Alberto', 'Martínez', 'Díaz', 'Cali', 'Calle 3', '3023456789', 'luis.alberto@example.com'),  (4,'4567890123', 'Ana', 'Patricia', 'Gómez', 'Pérez', 'Barranquilla', 'Calle 4', '3034567890', 'ana.patricia@example.com'),  (5,'5678901234', 'Carlos', 'Andrés', 'Hernández', 'Ortiz', 'Cartagena', 'Calle 5', '3045678901', 'carlos.andres@example.com'),  (6,'6789012345', 'Sofía', 'Andrea', 'López', 'Jiménez', 'Pereira', 'Calle 6', '3056789012', 'sofia.andrea@example.com'),  (7,'7890123456', 'Miguel', 'Ángel', 'Sánchez', 'Ramírez', 'Bucaramanga', 'Calle 7', '3067890123', 'miguel.angel@example.com'),  (8,'8901234567', 'Lucía', 'María', 'Ramírez', 'Torres', 'Santa Marta', 'Calle 8', '3078901234', 'lucia.maria@example.com'),  (9,'9012345678', 'José', 'Luis', 'Torres', 'Vargas', 'Manizales', 'Calle 9', '3089012345', 'jose.luis@example.com'),  (10,'1023456789', 'Elena', 'Isabel', 'Flores', 'Gómez', 'Ibagué', 'Calle 10', '3090123456', 'elena.isabel@example.com'),  (11,'1123456789', 'Andrés', 'Felipe', 'Moreno', 'López', 'Neiva', 'Calle 11', '3101234567', 'andres.felipe@example.com'),  (12,'1223456789', 'Paula', 'Andrea', 'Rojas', 'Martínez', 'Popayán', 'Calle 12', '3112345678', 'paula.andrea@example.com'),  (13,'1323456789', 'Fernando', 'José', 'Castro', 'Hernández', 'Pasto', 'Calle 13', '3123456789', 'fernando.jose@example.com'),  (14,'1423456789', 'Gloria', 'Elena', 'Vargas', 'Díaz', 'Sincelejo', 'Calle 14', '3134567890', 'gloria.elena@example.com'),  (15,'1523456789', 'Martín', 'Antonio', 'Méndez', 'Pérez', 'Tunja', 'Calle 15', '3145678901', 'martin.antonio@example.com'),  (16,'1623456789', 'Laura', 'Marcela', 'Cruz', 'Gómez', 'Armenia', 'Calle 16', '3156789012', 'laura.marcela@example.com'),  (17,'1723456789', 'Pedro', 'Juan', 'Ortiz', 'Ramírez', 'Riohacha', 'Calle 17', '3167890123', 'pedro.juan@example.com'),  (18,'1823456789', 'Natalia', 'Fernanda', 'Guerrero', 'López', 'Montería', 'Calle 18', '3178901234', 'natalia.fernanda@example.com'),  (19,'1923456789', 'Felipe', 'Andrés', 'Muñoz', 'Martínez', 'Villavicencio', 'Calle 19', '3189012345', 'felipe.andres@example.com'),  (20,'2023456789', 'Camila', 'Isabel', 'Suárez', 'Hernández', 'Cúcuta', 'Calle 20', '3190123456', 'camila.isabel@example.com'),  (21,'2123456789', 'Jorge', 'Luis', 'Jiménez', 'Torres', 'Valledupar', 'Calle 21', '3201234567', 'jorge.luis@example.com'),  (22,'2223456789', 'Patricia', 'María', 'Fernández', 'Vargas', 'Florencia', 'Calle 22', '3212345678', 'patricia.maria@example.com'),  (23,'2323456789', 'Ricardo', 'Alberto', 'Ruiz', 'Gómez', 'San Andrés', 'Calle 23', '3223456789', 'ricardo.alberto@example.com'),  (24,'2423456789', 'Adriana', 'Marcela', 'Delgado', 'Martínez', 'Leticia', 'Calle 24', '3234567890', 'adriana.marcela@example.com'),  (25,'2523456789', 'David', 'Alejandro', 'García', 'Pérez', 'Quibdó', 'Calle 25', '3245678901', 'david.alejandro@example.com'),  (26,'2623456789', 'Clara', 'Marcela', 'Ortiz', 'Vargas', 'Buga', 'Calle 26', '3256789012', 'clara.marcela@example.com'),  (27,'2723456789', 'Hugo', 'Alberto', 'González', 'Ramírez', 'Magangué', 'Calle 27', '3267890123', 'hugo.alberto@example.com'),  (28,'2823456789', 'Santiago', 'José', 'Pardo', 'Díaz', 'Buenaventura', 'Calle 28', '3278901234', 'santiago.jose@example.com'),  (29,'2923456789', 'Valentina', 'Isabel', 'Cardona', 'Hernández', 'Tumaco', 'Calle 29', '3289012345', 'valentina.isabel@example.com'),  (30,'3023456789', 'Alejandro', 'Antonio', 'Barrios', 'Vargas', 'Yopal', 'Calle 30', '3290123456', 'alejandro.antonio@example.com'),  (31,'3123456789', 'Gabriela', 'María', 'Navarro', 'Gómez', 'Girardot', 'Calle 31', '3301234567', 'gabriela.maria@example.com'),  (32,'3223456789', 'Sebastián', 'David', 'Vélez', 'Martínez', 'Sogamoso', 'Calle 32', '3312345678', 'sebastian.david@example.com'),  (33,'3323456789', 'Mónica', 'Andrea', 'Arévalo', 'Pérez', 'Ipiales', 'Calle 33', '3323456789', 'monica.andrea@example.com'),  (34,'3423456789', 'Javier', 'Francisco', 'Montoya', 'Ramírez', 'San José del Guaviare', 'Calle 34', '3334567890', 'javier.francisco@example.com'),  (35,'3523456789', 'Daniela', 'Fernanda', 'Espinosa', 'García', 'Puerto Asís', 'Calle 35', '3345678901', 'daniela.fernanda@example.com'),  (36,'3623456789', 'Álvaro', 'Enrique', 'Blanco', 'López', 'San Gil', 'Calle 36', '3356789012', 'alvaro.enrique@example.com'),  (37,'3723456789', 'Susana', 'Patricia', 'Lara', 'Martínez', 'El Carmen de Bolívar', 'Calle 37', '3367890123', 'susana.patricia@example.com'),  (38,'3823456789', 'Rodrigo', 'Andrés', 'Rangel', 'Vargas', 'Lorica', 'Calle 38', '3378901234', 'rodrigo.andres@example.com'),  (39,'3923456789', 'Tatiana', 'Isabel', 'Pineda', 'Hernández', 'Zipaquirá', 'Calle 39', '3389012345', 'tatiana.isabel@example.com'),  (40,'4023456789', 'Hernán', 'Antonio', 'Castaño', 'Pérez', 'Piedecuesta', 'Calle 40', '3390123456', 'hernan.antonio@example.com'); |

**Vehiculo**: Para ingresar datos en la tabla ‘vehiculo’, la cual muestra (idvehiculo, placa, modelo, motor, color, capacidad, puertas, tipo\_vehiculo, suroof, referencia), se emplea el siguiente comando:

|  |
| --- |
| INSERT INTO vehiculo (idvehiculo, placa, modelo, motor, color, capacidad, puertas, tipo\_vehiculo, suroof, referencia) VALUES |
| (1,'ABC123', 'Toyota Corolla', '1.8L', 'Rojo', '5', '4', 'Sedan', 'No', '2020'), |
| (2,'DEF456', 'Honda CR-V', '2.4L', 'Negro', '7', '5', 'SUV', 'Sí', '2021'),  (3,'GHI789', 'Ford Ranger', '3.2L', 'Blanco', '5', '4', 'Camioneta', 'No', '2019'),  (4,'JKL012', 'BMW Serie 4', '2.0L', 'Azul', '4', '2', 'Coupé', 'Sí', '2022'),  (5,'MNO345', 'Mazda MX-5', '2.0L', 'Gris', '2', '2', 'Convertible', 'Sí', '2018'),  (6,'PQR678', 'Volkswagen Golf', '1.4L', 'Verde', '5', '4', 'Hatchback', 'No', '2017'),  (7,'STU901', 'Chevrolet Silverado', '5.3L', 'Plateado', '5', '4', 'Pickup', 'No', '2020'),  (8,'VWX234', 'Chrysler Pacifica', '3.6L', 'Negro', '8', '5', 'Minivan', 'Sí', '2021'),  (9,'YZA567', 'Hyundai Elantra', '2.0L', 'Rojo', '5', '4', 'Sedan', 'No', '2019'),  (10,'BCD890', 'Kia Sorento', '2.5L', 'Blanco', '7', '5', 'SUV', 'Sí', '2020'),  (12,'EFG123', 'Nissan Frontier', '2.5L', 'Gris', '5', '4', 'Camioneta', 'No', '2021'),  (13,'HIJ456', 'Audi A5', '2.0L', 'Azul', '4', '2', 'Coupé', 'Sí', '2022'),  (14,'KLM789', 'Ford Mustang', '5.0L', 'Amarillo', '4', '2', 'Convertible', 'Sí', '2019'),  (15,'NOP012', 'Honda Fit', '1.5L', 'Verde', '5', '4', 'Hatchback', 'No', '2018'),  (16,'QRS345', 'Ram 1500', '5.7L', 'Plateado', '5', '4', 'Pickup', 'No', '2020'),  (17,'TUV678', 'Toyota Sienna', '3.5L', 'Negro', '8', '5', 'Minivan', 'Sí', '2021'),  (18,'WXY901', 'Mercedes-Benz Clase C', '2.0L', 'Rojo', '5', '4', 'Sedan', 'No', '2019'),  (19,'ZAB234', 'Subaru Outback', '2.5L', 'Blanco', '5', '5', 'SUV', 'Sí', '2020'),  (20,'CDE567', 'GMC Sierra', '5.3L', 'Gris', '5', '4', 'Camioneta', 'No', '2021'),  (21,'FGH890', 'Lexus RC', '3.5L', 'Azul', '4', '2', 'Coupé', 'Sí', '2022'),  (22,'IJK123', 'Chevrolet Camaro', '6.2L', 'Amarillo', '4', '2', 'Convertible', 'Sí', '2019'),  (23,'LMN456', 'Ford Fiesta', '1.6L', 'Verde', '5', '4', 'Hatchback', 'No', '2018'),  (24,'OPQ789', 'Toyota Tacoma', '3.5L', 'Plateado', '5', '4', 'Pickup', 'No', '2020'),  (25,'RST012', 'Honda Odyssey', '3.5L', 'Negro', '8', '5', 'Minivan', 'Sí', '2021'),  (26,'UVW345', 'Audi A4', '2.0L', 'Rojo', '5', '4', 'Sedan', 'No', '2019'),  (27,'XYZ678', 'Mazda CX-5', '2.5L', 'Blanco', '5', '5', 'SUV', 'Sí', '2020'),  (28,'ABC901', 'Ford F-150', '3.5L', 'Gris', '5', '4', 'Camioneta', 'No', '2021'),  (29,'DEF234', 'BMW M4', '3.0L', 'Azul', '4', '2', 'Coupé', 'Sí', '2022'),  (31,'GHI567', 'Porsche 911', '3.0L', 'Amarillo', '4', '2', 'Convertible', 'Sí', '2019'),  (32,'JKL890', 'Chevrolet Spark', '1.4L', 'Verde', '5', '4', 'Hatchback', 'No', '2018'),  (33,'MNO123', 'Honda Ridgeline', '3.5L', 'Plateado', '5', '4', 'Pickup', 'No', '2020'),  (34,'PQR456', 'Kia Carnival', '3.5L', 'Negro', '8', '5', 'Minivan', 'Sí', '2021'),  (35,'STU789', 'Lexus ES', '2.5L', 'Rojo', '5', '4', 'Sedan', 'No', '2019'),  (36,'VWX012', 'Jeep Grand Cherokee', '3.6L', 'Blanco', '5', '5', 'SUV', 'Sí', '2020'),  (37,'YZA345', 'Nissan Titan', '5.6L', 'Gris', '5', '4', 'Camioneta', 'No', '2021'),  (38,'BCD678', 'Mercedes-Benz Clase E', '3.0L', 'Azul', '4', '2', 'Coupé', 'Sí', '2022'),  (39,'EFG901', 'Jaguar F-Type', '2.0L', 'Amarillo', '4', '2', 'Convertible', 'Sí', '2019'),  (41,'HIJ234', 'Mini Cooper', '1.5L', 'Verde', '5', '4', 'Hatchback', 'No', '2018'),  (42,'KLM567', 'Chevrolet Colorado', '2.5L', 'Plateado', '5', '4', 'Pickup', 'No', '2020'),  (43,'NOP890', 'Hyundai Palisade', '3.8L', 'Negro', '8', '5', 'Minivan', 'Sí', '2021'),  (44,'QRS123', 'Volvo XC90', '2.0L', 'Azul', '7', '5', 'SUV', 'Sí', '2021'),  (45,'TUV234', 'Tesla Model S', 'Eléctrico', 'Blanco', '5', '4', 'Sedan', 'Sí', '2022'),  (46,'WXY345', 'Audi Q5', '2.0L', 'Rojo', '5', '5', 'SUV', 'Sí', '2020'),  (47,'ZAB456', 'Mitsubishi Outlander', '2.4L', 'Negro', '7', '5', 'SUV', 'Sí', '2021'),  (48,'CDE567', 'Toyota Hilux', '2.8L', 'Blanco', '5', '4', 'Camioneta', 'No', '2020'),  (49,'FGH678', 'Honda Pilot', '3.5L', 'Gris', '8', '5', 'SUV', 'Sí', '2021'),  (50,'IJK789', 'Mercedes-Benz GLC', '2.0L', 'Azul', '5', '5', 'SUV', 'Sí', '2022'); |

**Alquiler**: Para ingresar datos en la tabla ‘area’, la cual muestra el id del parque, nombre y extensión del área específica del parque, se emplea el siguiente comando:

|  |
| --- |
| INSERT INTO alquiler (idalquiler, Fecha\_salida, Fecha\_esperada, Fecha\_entrega, Sucursal\_salida, Sucursal\_entrega, valor\_dia, valor\_semanal, valor\_cotizado, descuento, sucursal\_idsucursal, Cliente\_idcliente, Vehiculo\_idvehiculo, Empleado\_idempleado) VALUES |
| (1,'2023-01-01', '2023-01-08', '2023-01-07', 'Bogotá', 'Cali', 50000, 300000, 285000, 15000, 1, 1, 1, 1), |
| (2,'2023-01-02', '2023-01-09', '2023-01-08', 'Medellín', 'Cali', 60000, 350000, 330000, 20000, 2, 2, 2, 2), |
| (3,'2023-01-03', '2023-01-10', '2023-01-09', 'Medellín', 'Cali', 70000, 400000, 380000, 20000, 2, 3, 3, 3),  (4,'2023-01-04', '2023-01-11', '2023-01-10', 'Cartagena', 'Barranquilla', 80000, 450000, 430000, 20000, 5, 4, 4, 4),  (5,'2023-01-05', '2023-01-12', '2023-01-11', 'Barranquilla', 'Cartagena', 90000, 500000, 480000, 20000, 4, 5, 5, 5),  (6,'2023-01-06', '2023-01-13', '2023-01-12', 'Bogotá', 'Cali', 55000, 320000, 310000, 10000, 1, 6, 6, 6),  (7,'2023-01-07', '2023-01-14', '2023-01-13', 'Medellín', 'Barranquilla', 65000, 370000, 360000, 10000, 2, 7, 7, 7),  (8,'2023-01-08', '2023-01-15', '2023-01-14', 'Cali', 'Bogotá', 75000, 420000, 410000, 10000, 3, 8, 8, 8),  (9,'2023-01-09', '2023-01-16', '2023-01-15', 'Barranquilla', 'Medellín', 85000, 470000, 460000, 10000, 4, 9, 9, 9),  (10,'2023-01-10', '2023-01-17', '2023-01-16', 'Cartagena', 'Medellín', 95000, 520000, 510000, 10000, 5, 10, 10, 10),  (12,'2023-01-11', '2023-01-18', '2023-01-17', 'Bogotá', 'Barranquilla', 60000, 330000, 320000, 10000, 1, 11, 10, 11),  (13,'2023-01-12', '2023-01-19', '2023-01-18', 'Medellín', 'Cali', 70000, 380000, 370000, 10000, 2, 12, 12, 12),  (14,'2023-01-13', '2023-01-20', '2023-01-19', 'Cali', 'Barranquilla', 80000, 430000, 420000, 10000, 3, 13, 13, 13),  (15,'2023-01-14', '2023-01-21', '2023-01-20', 'Barranquilla', 'Cartagena', 90000, 480000, 470000, 10000, 4, 14, 14, 14),  (16,'2023-01-15', '2023-01-22', '2023-01-21', 'Cartagena', 'Bogotá', 100000, 530000, 520000, 10000, 5, 15, 15, 15),  (17,'2023-01-16', '2023-01-23', '2023-01-22', 'Medellín', 'Bogotá', 65000, 350000, 340000, 10000, 2, 16, 16, 16),  (18,'2023-01-17', '2023-01-24', '2023-01-23', 'Bogotá', 'Medellín', 75000, 400000, 390000, 10000, 1, 17, 17, 17),  (19,'2023-01-18', '2023-01-25', '2023-01-24', 'Cali', 'Cartagena', 85000, 450000, 440000, 10000, 3, 18, 18, 18),  (20,'2023-01-31', '2023-02-07', '2023-02-06', 'Bogotá', 'Bogotá', 80000, 410000, 400000, 10000, 1, 31, 31, 12),  (21,'2023-01-19', '2023-01-26', '2023-01-25', 'Barranquilla', 'Cali', 95000, 500000, 490000, 10000, 4, 19, 19, 19),  (22,'2023-01-20', '2023-01-27', '2023-01-26', 'Cartagena', 'Medellín', 105000, 550000, 540000, 10000, 5, 20, 20, 20),  (23,'2023-01-21', '2023-01-28', '2023-01-27', 'Bogotá', 'Bogotá', 70000, 370000, 360000, 10000, 1, 21, 21, 21),  (24,'2023-01-22', '2023-01-29', '2023-01-28', 'Medellín', 'Medellín', 80000, 420000, 410000, 10000, 2, 22, 22, 10),  (25,'2023-01-23', '2023-01-30', '2023-01-29', 'Cali', 'Bogotá', 90000, 470000, 460000, 10000, 3, 23, 23, 8),  (26,'2023-01-24', '2023-01-31', '2023-01-30', 'Barranquilla', 'Cartagena', 100000, 520000, 510000, 10000, 4, 24, 24, 7),  (27,'2023-01-25', '2023-02-01', '2023-01-31', 'Cartagena', 'Bogotá', 110000, 570000, 560000, 10000, 5, 25, 25, 25),  (28,'2023-01-26', '2023-02-02', '2023-02-01', 'Bogotá', 'Medellín', 75000, 390000, 380000, 10000, 1, 26, 26, 22),  (29,'2023-01-27', '2023-02-03', '2023-02-02', 'Cali', 'Medellín', 85000, 440000, 430000, 10000, 3, 27, 27, 4),  (30,'2023-01-30', '2023-02-06', '2023-02-05', 'Cartagena', 'Medellín', 115000, 590000, 580000, 10000, 5, 30, 31, 10),  (31,'2023-01-28', '2023-02-04', '2023-02-03', 'Medellín', 'Cali', 95000, 490000, 480000, 10000, 2, 28, 28, 20),  (32,'2023-01-29', '2023-02-05', '2023-02-04', 'Barranquilla', 'Bogotá', 105000, 540000, 530000, 10000, 4, 29, 29, 25); |

### Consultas de Datos

Las consultas en una base de datos son indispensables, ya que facilitan el acceso y la recuperación de información almacenada. Además, permiten mantener la base de datos actualizada mediante la inserción, modificación y actualización de datos. Son fundamentales para almacenar, manipular y recuperar datos de manera eficiente y segura.

### Descripción

**25 CONSULTAS:**

1. cuantos alquileres a tenido el cliente con el identidicador 10

SELECT COUNT(\*) AS total\_alquileres FROM alquiler WHERE Cliente\_idcliente = 10;

2. cuantos carros rojos hay

SELECT COUNT(\*) AS total\_carros\_rojos FROM vehiculo WHERE color = 'Rojo';

3. cuantos carros han salido de la sucursal '1 medellin' a la sucursal '4 barranquilla'

SELECT COUNT(\*) AS total\_carros FROM alquiler WHERE Sucursal\_salida = 'Medellín' AND Sucursal\_entrega = 'Barranquilla';

4. cual ha sido el alquiler mas costoso

SELECT \* FROM alquiler ORDER BY valor\_cotizado DESC LIMIT 1;

5. cual es el carro con mas alquileres

SELECT Vehiculo\_idvehiculo, COUNT(\*) AS total\_alquileres FROM alquiler GROUP BY Vehiculo\_idvehiculo ORDER BY total\_alquileres DESC LIMIT 1;

6. devuelve las fechas en las que se alquilo el vehiculo con id 10

SELECT Fecha\_salida, Fecha\_esperada, Fecha\_entrega FROM alquiler WHERE Vehiculo\_idvehiculo = 10;

7. cuel es la sucursal con mas alquileres

SELECT Sucursal\_salida, COUNT(\*) AS total\_alquileres FROM alquiler GROUP BY Sucursal\_salida ORDER BY total\_alquileres DESC LIMIT 1;

8. cual es el mes con más alquileres

SELECT MONTH(Fecha\_salida) AS mes, COUNT(\*) AS total\_alquileres FROM alquiler GROUP BY mes ORDER BY total\_alquileres DESC LIMIT 1;

9. cuantas veces un cliente a alquilado el mismo vehiculo

SELECT Cliente\_idcliente, Vehiculo\_idvehiculo, COUNT(\*) AS total\_alquileres FROM alquiler

GROUP BY Cliente\_idcliente, Vehiculo\_idvehiculo HAVING COUNT(\*) > 1;

10. cual es el venderor que mas alquileres a registrado

SELECT Empleado\_idempleado, COUNT(\*) AS total\_alquileres FROM alquiler

GROUP BY Empleado\_idempleado ORDER BY total\_alquileres DESC LIMIT 1;

11. cuantos alquileres se han registrado en la sucursal de medellin

SELECT COUNT(\*) AS total\_alquileres FROM alquiler

WHERE Sucursal\_salida = 'Medellín';

12. devuelve el total de ingresos por alquiler

SELECT SUM(valor\_cotizado) AS total\_ingresos FROM alquiler;

13. Listar todos los vehículos que nunca han sido alquilados

SELECT v.idvehiculo FROM vehiculo v

LEFT JOIN alquiler a ON v.idvehiculo = a.Vehiculo\_idvehiculo WHERE a.Vehiculo\_idvehiculo IS NULL;

14. Calcular el promedio de días de alquiler por vehículo

SELECT Vehiculo\_idvehiculo, AVG(DATEDIFF(Fecha\_entrega, Fecha\_salida)) AS promedio\_dias

FROM alquiler GROUP BY Vehiculo\_idvehiculo;

15. vehículo con el mayor ingreso total

SELECT Vehiculo\_idvehiculo, SUM(valor\_cotizado) AS total\_ingresos FROM alquiler

GROUP BY Vehiculo\_idvehiculo ORDER BY total\_ingresos DESC LIMIT 1;

16. empleados que han registrado alquileres en más de una sucursal

SELECT Empleado\_idempleado, COUNT(DISTINCT Sucursal\_salida) AS total\_sucursales FROM alquiler

GROUP BY Empleado\_idempleado HAVING total\_sucursales > 1;

17. total de descuentos aplicados en los alquileres

SELECT SUM(descuento) AS total\_descuentos FROM alquiler;

18. cliente que ha pagado más en alquileres

SELECT Cliente\_idcliente, SUM(valor\_pagado) AS total\_pagado FROM alquiler

GROUP BY Cliente\_idcliente ORDER BY total\_pagado DESC LIMIT 1;

19. vehículos que han sido alquilados en más de una sucursal:

SELECT Vehiculo\_idvehiculo, COUNT(DISTINCT Sucursal\_salida) AS total\_sucursales FROM alquiler

GROUP BY Vehiculo\_idvehiculo HAVING total\_sucursales > 1;

20. alquileres con una duración superior a 7 días

SELECT \* FROM alquiler WHERE DATEDIFF(Fecha\_entrega, Fecha\_salida) > 7;

21. vehículos que han sido alquilados en el último año

SELECT DISTINCT Vehiculo\_idvehiculo FROM alquiler

WHERE Fecha\_salida >= DATE\_SUB(CURDATE(), INTERVAL 1 YEAR);

22. promedio de días que un vehículo ha estado alquilado

SELECT AVG(DATEDIFF(Fecha\_entrega, Fecha\_salida)) AS promedio\_dias\_alquilado FROM alquiler;

23. vehículos que han sido alquilados en los últimos seis meses

SELECT DISTINCT Vehiculo\_idvehiculo FROM alquiler

WHERE Fecha\_salida >= DATE\_SUB(CURDATE(), INTERVAL 6 MONTH);

24. fechas con el menor número de alquileres

SELECT Fecha\_salida, COUNT(\*) AS total\_alquileres FROM alquiler

GROUP BY Fecha\_salida ORDER BY total\_alquileres ASC LIMIT 1;

25. lista de todos los empleados y cuántos alquileres han registrado

SELECT Empleado\_idempleado, COUNT(\*) AS total\_alquileres

FROM alquiler GROUP BY Empleado\_idempleado;

**5 FUNCIONES:**

1. **Calcula el total de ingresos generados por una sucursal en especial.**

|  |
| --- |
| c |
|  |
| CREATE FUNCTION TotalIngresosSucursal(sucursal VARCHAR(45))  RETURNS DECIMAL(10, 2)  BEGIN  DECLARE total\_ingresos DECIMAL(10, 2);  SELECT SUM(valor\_cotizado - descuento) INTO total\_ingresos  FROM alquiler  WHERE Sucursal\_salida = sucursal;  RETURN total\_ingresos;  END //  DELIMITER ; |

1. Calcular el promedio de días de alquiler de un vehículo específico.

|  |
| --- |
| DELIMITER // |
|  |
| CREATE FUNCTION PromedioDiasAlquilerVehiculo(vehiculo\_id INT)  RETURNS DECIMAL (10, 2)  BEGIN  DECLARE promedio\_dias DECIMAL (10, 2);  SELECT AVG(DATEDIFF(Fecha\_entrega, Fecha\_salida)) INTO promedio\_dias  FROM alquiler  WHERE Vehiculo\_idvehiculo = vehiculo\_id;  RETURN total\_alquileres;  END  DELIMITER ; |

1. Devuelve el cliente que ha pagado más en alquileres.

|  |
| --- |
| DELIMITER // |
|  |
| CREATE FUNCTION ClienteMaximoPagado()  RETURNS INT  BEGIN  DECLARE cliente\_id INT;  SELECT Cliente\_idcliente INTO cliente\_id  FROM alquiler  GROUP BY Cliente\_idcliente  ORDER BY SUM(valor\_pagado) DESC  LIMIT 1;  RETURN cliente\_id;  END //  DELIMITER ; |

1. Calcula el total de descuentos aplicados en un mes específico

|  |
| --- |
| DELIMITER // |
|  |
| CREATE FUNCTION TotalDescuentosMes(mes INT)  RETURNS DECIMAL (10, 2)  BEGIN  DECLARE total\_descuentos DECIMAL (10, 2);  SELECT SUM(descuento) INTO total\_descuentos  FROM alquiler  WHERE MONTH(Fecha\_salida) = mes;  RETURN total\_descuentos;  END //  DELIMITER ; |

1. Cuenta cuántos alquileres ha realizado un cliente específico en un año determinado

|  |
| --- |
| DELIMITER // |
|  |
| CREATE FUNCTION AlquileresClienteAnio(cliente\_id INT, anio  RETURNS INT  BEGIN  DECLARE total\_alquileres INT;  SELECT COUNT(\*) INTO total\_alquileres  FROM alquiler  WHERE Cliente\_idcliente = cliente\_id AND YEAR(Fecha\_salida) = anio;  RETURN total\_alquileres;  END  DELIMITER ; |