AUTORENTAL

# 

**WILMER ALEXANDER ROJAS FUENTES**

**T1**

**PEDRO FELIPE GÓMEZ BONILLA**

**CAMPUSLANDS**

**SANDBOX  
RUTA JAVA**

**TIBU**

**2024**

Tabla de Contenidos

[**Introducción 4**](#_xy776j9pw1un)

[**Caso de Estudio 5**](#_x5wjmxr9s7lx)

[**Planificación 5**](#_8muzr1pu3mlz)

[Construcción del Modelo Conceptual 5](#_akgvn7l12xs2)

[Descripción 5](#_qxuss0va0z8)

[Gráfica 5](#_u45t31dnlow2)

[Descripción Técnica 6](#_gww41npptmyg)

[Construcción del Modelo Lógico 6](#_nzoi4crbe63p)

[Descripción 6](#_46z7wrz43fjb)

[Gráfica 6](#_cx639fevpp57)

[Descripción Técnica 6](#_vemj55jop1dw)

[Normalización del Modelo Lógico 6](#_j975ki11pvw9)

[Primera Forma Normal (1FN) 7](#_ipsnvw5v2v0u)

[Descripción 7](#_yudex7qmv5l6)

[Gráfica 7](#_u1g2e96ymz30)

[Descripción Técnica 7](#_rjmpapoxneod)

[Segunda Forma Normal (2FN) 7](#_3fyocgo001fx)

[Descripción 8](#_n31b5bkefr23)

[Gráfica 8](#_mywvu5xf7mjo)

[Descripción Técnica 8](#_i9ctpni5mu37)

[Tercera Forma Normal (3FN) 8](#_aeip5fb7t3s5)

[Descripción 8](#_v6plqfy3484u)

[Gráfica 8](#_bcqznd6b310i)

[Descripción Técnica 8](#_qaz530fwxr49)

[Construcción del Modelo Físico 9](#_muq8ro9qbje6)

[Descripción 9](#_5g32igevz4r8)

[Código 9](#_ryjwvstap29b)

[Descripción Técnica 10](#_f9n2yx58hlja)

[Diagrama E-R 10](#_l2vl1lqjv6nt)

[Descripción 10](#_b542o9532ses)

[Gráfica 10](#_8wg77tt5rsnt)

[Descripción Técnica 10](#_whj2rpgnzs4v)

[Tablas 11](#_s3pvkylovom)

[Descripción 11](#_ye40r8z6ajh3)

[Gráfica 11](#_mgogeuideyfr)

[Descripción Técnica 11](#_4tau2idq5j2j)

[Relaciones entre Tablas 11](#_v02u5svki3o6)

[Descripción 11](#_mn4p1kzcs5d)

[Gráfica 12](#_ahrg4l6677cx)

[Descripción Técnica 12](#_v3u2pmnyvhxt)

[Inserción de Datos 12](#_khdudtfa7c4u)

[Descripción 12](#_r9p210nyg848)

[Gráfica 12](#_wpdzy6svxf2h)

[Descripción Técnica 12](#_cudxay80qe23)

[**Referencias 13**](#_ni6ac3h1zonf)

# 

# Introducción

El objetivo de este documento es guiar la creación y diseño de una base de datos para el concesionario de alquiler de vehículos **AutoRental**. Esta empresa ofrece vehículos en alquiler a los usuarios, quienes pueden recoger el vehículo en una sucursal y devolverlo en otra. Actualmente, **AutoRental** cuenta con 5 sucursales y planea expandirse a otras ciudades del país.

El valor del alquiler depende de la categoría del vehículo (sedán, compacto, camioneta, lujo, deportivo, etc.) y se cobra por días o semanas. Además, si un cliente no entrega el vehículo en la fecha pactada, se aplicará un recargo del 8% adicional por cada día de retraso.

# 

# 

# Caso de Estudio

En este caso de estudio, se desarrollará una base de datos que permita cumplir con los requisitos del sistema de alquiler de vehículos. La base de datos debe registrar el alquiler de un vehículo por parte de un cliente, incluyendo la salida desde una sucursal y su posterior ingreso a la sucursal de destino. Además, debe gestionar la fecha de alquiler y de entrega (en días y/o semanas), aplicar cargos adicionales por retraso en la entrega y clasificar los vehículos según su tipo (sedán, compacto, camioneta platón, camioneta de lujo, deportivo, etc.).

Las tablas que se crearán en este caso de estudio son las siguientes:

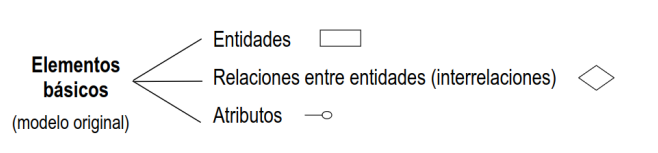
* Sucursales
* Empleados
* Clientes
* Vehículos
* Alquileres

# Planificación

## Construcción del Modelo Conceptual

En este modelo se procede a diseñar el modelo conceptual identificando las entidades, atributos y relaciones entre ellas. El mismo permitirá una visión clara y estructurada de la organización y conexión de los distintos elementos de la base de datos.

Para comprender el modelo conceptual, se tendrá en cuenta los elementos básicos del mismo.



### Descripción

**Entidades y atributos**

En este modelo se logra hacer las relaciones que tienen todas las entidades sobre cómo debe funcionar la base de datos para así proceder luego a crear el modelo lógico.

### **Clientes**

* **id**: Identificador único del cliente (clave primaria).
* **cedula**: Número de identificación único del cliente.
* **nombres**: Nombre(s) del cliente.
* **apellidos**: Apellido(s) del cliente.
* **ciudad\_residencia**: Ciudad en la que reside el cliente.
* **direccion**: Dirección detallada de residencia del cliente.
* **celular**: Número de celular del cliente.
* **correo**: Dirección de correo electrónico del cliente.

### **Empleados**

* **id**: Identificador único del empleado (clave primaria).
* **cedula**: Número de identificación único del empleado.
* **nombres**: Nombre(s) del empleado.
* **apellidos**: Apellido(s) del empleado.
* **direccion**: Dirección detallada de residencia del empleado.
* **ciudad\_residencia**: Ciudad en la que reside el empleado.
* **celular**: Número de celular del empleado.
* **e-mail**: Dirección de correo electrónico del empleado.
* **id\_sucursal**: Identificador de la sucursal donde trabaja el empleado (clave foránea a sucursales).

### **Sucursales**

* **id**: Identificador único de la sucursal (clave primaria).
* **ciudad**: Ciudad en la que se encuentra la sucursal.
* **direccion**: Dirección física de la sucursal.
* **celular**: Número de celular de contacto de la sucursal.
* **telefono\_fijo**: Número de teléfono fijo de contacto de la sucursal.
* **correo**: Dirección de correo electrónico de la sucursal.

### **Descuentos**

* **id**: Identificador único del descuento (clave primaria).
* **fecha\_inicio**: Fecha de inicio de la validez del descuento.
* **fecha\_fin**: Fecha de fin de la validez del descuento.
* **porcentaje\_descuento**: Porcentaje de descuento aplicable.

### **Tipo\_vehiculo**

* **id**: Identificador único del tipo de vehículo (clave primaria).
* **tipo\_vehiculo**: Nombre del tipo de vehículo (e.g., sedán, camioneta platón, deportivo).
* **id\_descuento**: Identificador de descuento aplicable a este tipo de vehículo (clave foránea a descuentos).

### **Vehiculos**

* **id**: Identificador único del vehículo (clave primaria).
* **placa**: Número de matrícula o placa del vehículo.
* **referencia**: Modelo o referencia del vehículo.
* **capacidad**: Capacidad de pasajeros o carga del vehículo.
* **sunroof**: Indicador de si el vehículo tiene sunroof (sí/no).
* **puertas**: Número de puertas del vehículo.
* **modelo**: Año de fabricación del vehículo.
* **motor**: Tipo o especificaciones del motor del vehículo.
* **color**: Color del vehículo.
* **id\_tipo\_vehiculo**: Identificador del tipo de vehículo (clave foránea a tipo\_vehiculo).

### **Alquileres**

* **id**: Identificador único del alquiler (clave primaria).
* **alquiler\_semana**: Costo de alquiler por semana.
* **alquiler\_dia**: Costo de alquiler por día.
* **fecha\_salida**: Fecha en la que el vehículo es retirado de la sucursal.
* **fecha\_esperada**: Fecha esperada de devolución del vehículo.
* **fecha\_llegada**: Fecha en la que el vehículo es entregado en la sucursal de destino.
* **valor\_cotizado**: Valor total calculado del alquiler, considerando el tipo de vehículo, duración, y descuentos aplicados.
* **valor\_pagado**: Valor que el cliente pagó por el alquiler.
* **id\_cliente**: Identificador del cliente que realiza el alquiler (clave foránea a clientes).
* **id\_empleado**: Identificador del empleado que procesó el alquiler (clave foránea a empleados).
* **id\_vehiculo**: Identificador del vehículo alquilado (clave foránea a vehiculos).
* **id\_sucursal\_salida**: Identificador de la sucursal de donde se retira el vehículo (clave foránea a sucursales).
* **id\_sucursal\_llegada**: Identificador de la sucursal donde se entrega el vehículo (clave foránea a sucursales).
* **id\_descuento**: Identificador del descuento aplicado al alquiler (clave foránea a descuentos).

### **Retraso**

* **id**: Identificador único para el retraso (clave primaria).
* **dias\_retraso**: Número de días de retraso en la entrega del vehículo.
* **%\_dias**: Porcentaje adicional que se cobra por cada día de retraso.
* **id\_alquiler**: Identificador del alquiler asociado a este retraso (clave foránea a alquileres).

**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:

Aquí tienes las relaciones y cardinalidades completas para las entidades que mencionaste en el modelo conceptual:

### **clientes - alquileres**

* **Relación**: “alquila”
* **Descripción**: Un cliente puede alquilar muchos vehículos, y un vehículo puede ser alquilado por muchos clientes a lo largo del tiempo.
* **Cardinalidad**: **1**(uno a muchos). Un cliente puede tener múltiples alquileres, pero cada alquiler está asociado a un único cliente.

### **empleados - alquileres**

* **Relación**: “verifica”
* **Descripción**: Un empleado puede verificar y procesar muchos alquileres, pero cada alquiler es verificado por un solo empleado.
* **Cardinalidad**: 1:N(uno a muchos). Un empleado puede estar asociado a muchos alquileres, pero cada alquiler tiene un solo empleado encargado de verificarlo.

### **sucursales - alquileres**

* **Relación:** “gestionan”
* **Descripción**: Una sucursal puede gestionar muchos alquileres (tanto de salida como de llegada), pero cada alquiler está asociado a una sucursal de salida y una sucursal de llegada.
* **Cardinalidad**: 1:N  
  (uno a muchos). Una sucursal puede gestionar múltiples alquileres, pero cada alquiler está vinculado a una sola sucursal de salida y una sola sucursal de llegada.

### **descuentos - alquileres**

* **Relación:** “aplica”
* Descripción: Un descuento puede ser aplicado a muchos alquileres, pero cada alquiler puede tener solo un descuento.
* Cardinalidad: 1:N  
  (uno a muchos). Un descuento puede aplicarse a varios alquileres, pero cada alquiler tiene un solo descuento.

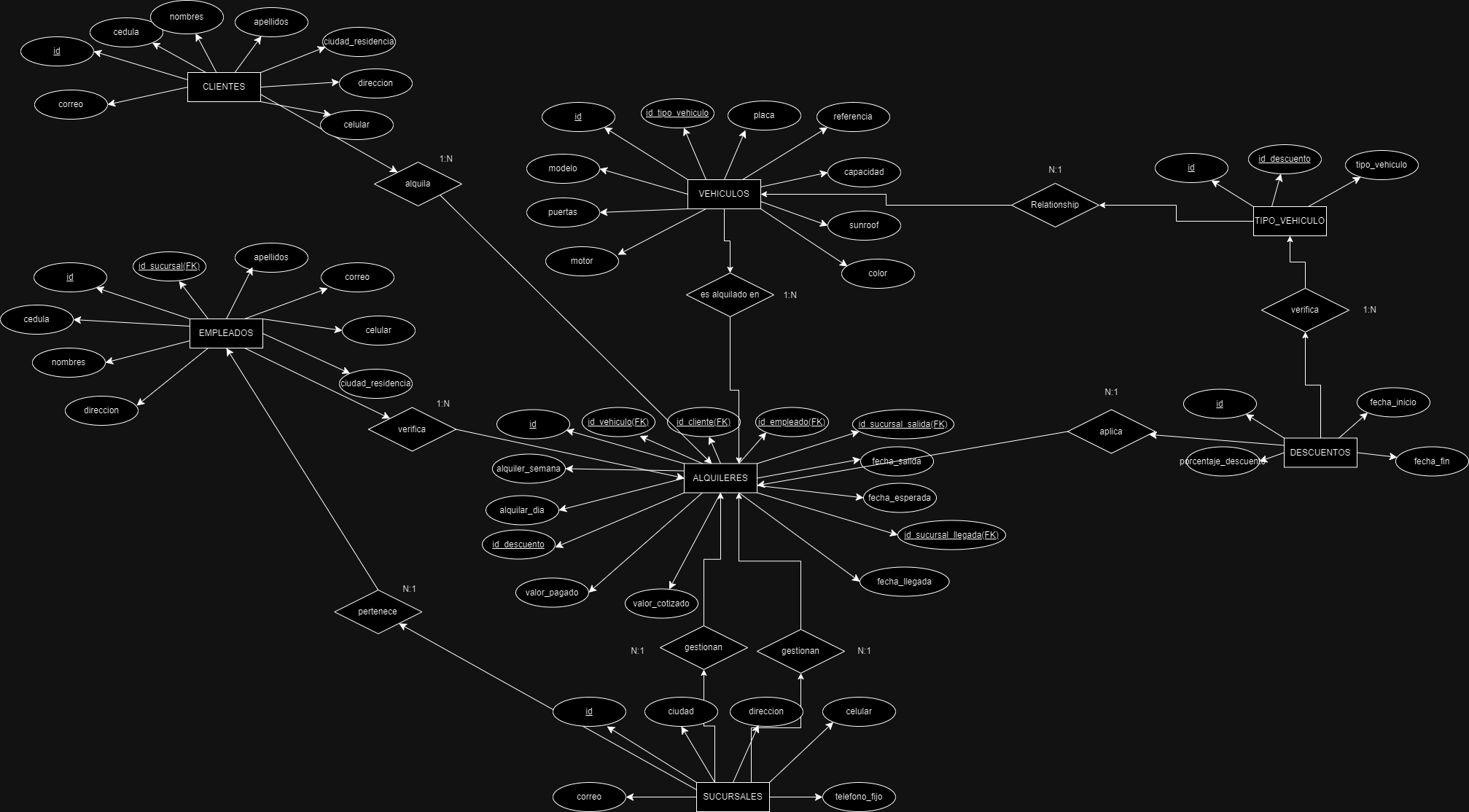
### **vehículos - alquileres**

* **Relación:** “es alquilado en”
* Descripción: Un vehículo puede ser alquilado muchas veces, pero cada alquiler está vinculado a un solo vehículo.
* Cardinalidad: 1:N  
  (uno a muchos). Un vehículo puede tener varios alquileres, pero cada alquiler está asociado a un único vehículo.

**tipo\_vehiculo - vehículos**

* **Relación**: "clasifica"
* **Descripción**: Un tipo de vehículo puede clasificar muchos vehículos, pero cada vehículo está asociado a un solo tipo de vehículo.
* **Cardinalidad**: 1:N  
  (uno a muchos). Un tipo de vehículo puede estar asociado con varios vehículos, pero cada vehículo tiene un solo tipo de vehículo.

### 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**

**Clientes**:

* **id**: INT PRIMARY KEY.
* **cedula**: VARCHAR(45) NOT NULL.
* **nombres**: VARCHAR(45) NOT NULL.
* **apellidos**: VARCHAR(45) NOT NULL.
* **ciudad\_residencia**: VARCHAR(45) NOT NULL.
* **dirección**: VARCHAR(200) NOT NULL.
* **celular**: VARCHAR(45) NOT NULL.
* **correo**: VARCHAR(45) NOT NULL.

**Empleados**:

* **id**: INT PRIMARY KEY.
* **cedula**: VARCHAR(45) NOT NULL.
* **nombres**: VARCHAR(45) NOT NULL.
* **apellidos**: VARCHAR(45) NOT NULL.
* **direccion**: VARCHAR(200) NOT NULL.
* **ciudad\_residencia**: VARCHAR(45) NOT NULL.
* **celular**: VARCHAR(45) NOT NULL.
* **correo**: VARCHAR(45) NOT NULL.
* **id\_sucursal**: INT NOT NULL, FOREIGN KEY (id\_sucursal) REFERENCES sucursales (id).

**Sucursales**:

* **id**: INT PRIMARY KEY.
* **ciudad**: VARCHAR(45) NOT NULL.
* **direccion**: VARCHAR(200) NOT NULL.
* **celular**: VARCHAR(45) NOT NULL.
* **telefono\_fijo**: VARCHAR(45) NOT NULL.
* **correo**: VARCHAR(45) NOT NULL.

**Descuentos**:

* **id**: INT PRIMARY KEY.
* **fecha\_inicio**: DATE NOT NULL.
* **fecha\_fin**: DATE NOT NULL.
* **porcentaje\_descuento**: INT NOT NULL.

**Tipo\_vehiculo**:

* **id**: INT PRIMARY KEY.
* **tipo\_vehiculo**: VARCHAR(45) NOT NULL.
* **id\_descuento**: INT NOT NULL, FOREIGN KEY (id\_descuento) REFERENCES Descuentos(id).

**Vehículos**:

* **id**: INT PRIMARY KEY.
* **placa**: VARCHAR(45) NOT NULL.
* **referencia**: VARCHAR(45) NOT NULL.
* **capacidad**: INT NOT NULL.
* **sunroof**: VARCHAR(45) NOT NULL.
* **puertas**: VARCHAR(45) NOT NULL.
* **modelo**: VARCHAR(45) NOT NULL.
* **motor**: VARCHAR(45) NOT NULL.
* **color**: VARCHAR(45) NOT NULL.
* **id\_tipo\_vehiculo**: INT NOT NULL, FOREIGN KEY (id\_tipo\_vehiculo) REFERENCES Tipo\_vehiculo(id).

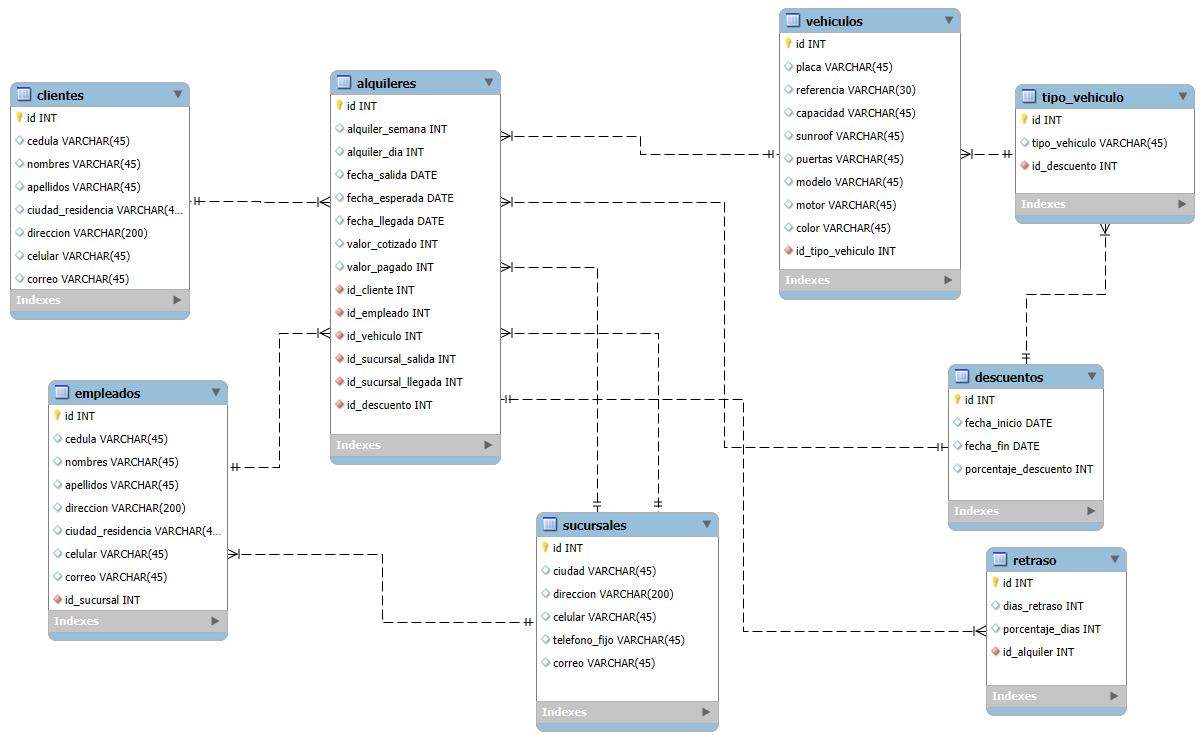
**Alquileres**:

* **id**: INT PRIMARY KEY.
* **alquiler\_semana**: INT NOT NULL.
* **alquiler\_dia**: INT NOT NULL.
* **fecha\_salida**: DATE NOT NULL.
* **fecha\_esperada**: DATE NOT NULL.
* **fecha\_llegada**: DATE NOT NULL.
* **valor\_cotizado**: INT NOT NULL.
* **valor\_pagado**:INT NOT NULL.
* **id\_cliente**: INT NOT NULL, FOREIGN KEY (id\_cliente) REFERENCES Clientes(id).
* **id\_empleado**: INT NOT NULL, FOREIGN KEY (id\_empleado) REFERENCES Empleados(id).
* **id\_vehiculo**: INT NOT NULL, FOREIGN KEY (id\_vehiculo) REFERENCES Vehículos(id).
* **id\_sucursal\_salida**: INT NOT NULL, FOREIGN KEY (id\_sucursal\_salida) REFERENCES Sucursales(id).
* **id\_sucursal\_llegada**: INT NOT NULL, FOREIGN KEY (id\_sucursal\_llegada) REFERENCES Sucursales(id).
* **id\_descuento**: INT NOT NULL, FOREIGN KEY (id\_descuento) REFERENCES Descuentos(id).

**Retraso**:

* **id**: INT PRIMARY KEY.
* **dias\_retraso**: INT NOT NULL.
* **porcentaje\_dias**: INT NOT NULL.
* **id\_alquiler**: INT NOT NULL, FOREIGN KEY (id\_alquiler) REFERENCES Alquileres(id).

### Gráfica



## Normalización del Modelo Lógico

Normalización del Modelo Lógico

Se realizó el proceso de la normalización de las tablas anteriormente visualizadas para

organizar los datos de manera más eficiente,minimizando redundancias y dependencias

transitivas en la base de datos en desarrollo.

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.

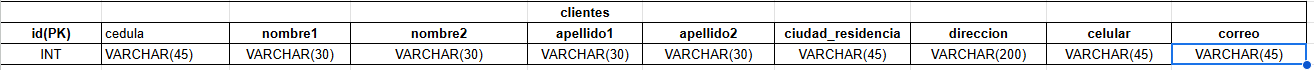
Descripción

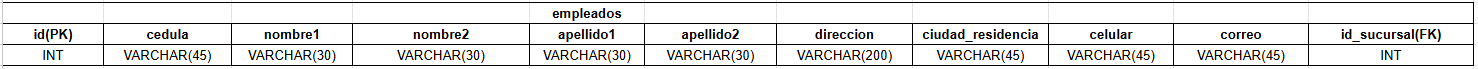
La primera forma normal, es el primer nivel de normalización en el diseño de la base de datos

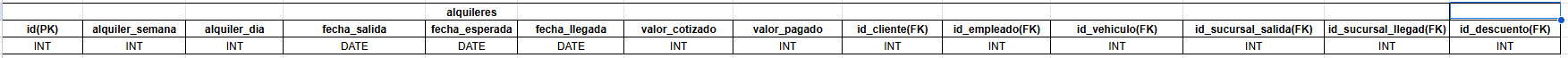
que se aplicará a las tablas de la base de datos para garantizar la organización de los datos de

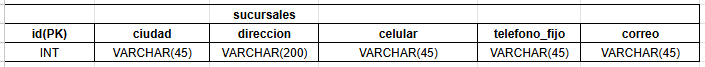
manera que evite redundancias y asegure la consistencia de la información.

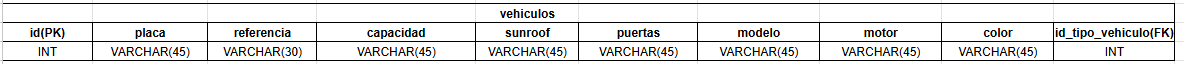
#### Gráfica

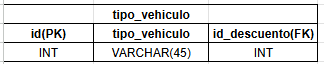


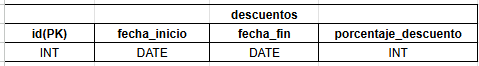


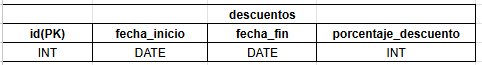


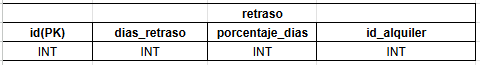












#### **Segunda Forma Normal (2FN)**

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

#### ❖ Está en 1FN.

#### ❖ Todos los atributos no clave (no pertenecientes a una clave primaria compuesta)

#### dependen completamente de la clave primaria.

#### Descripción

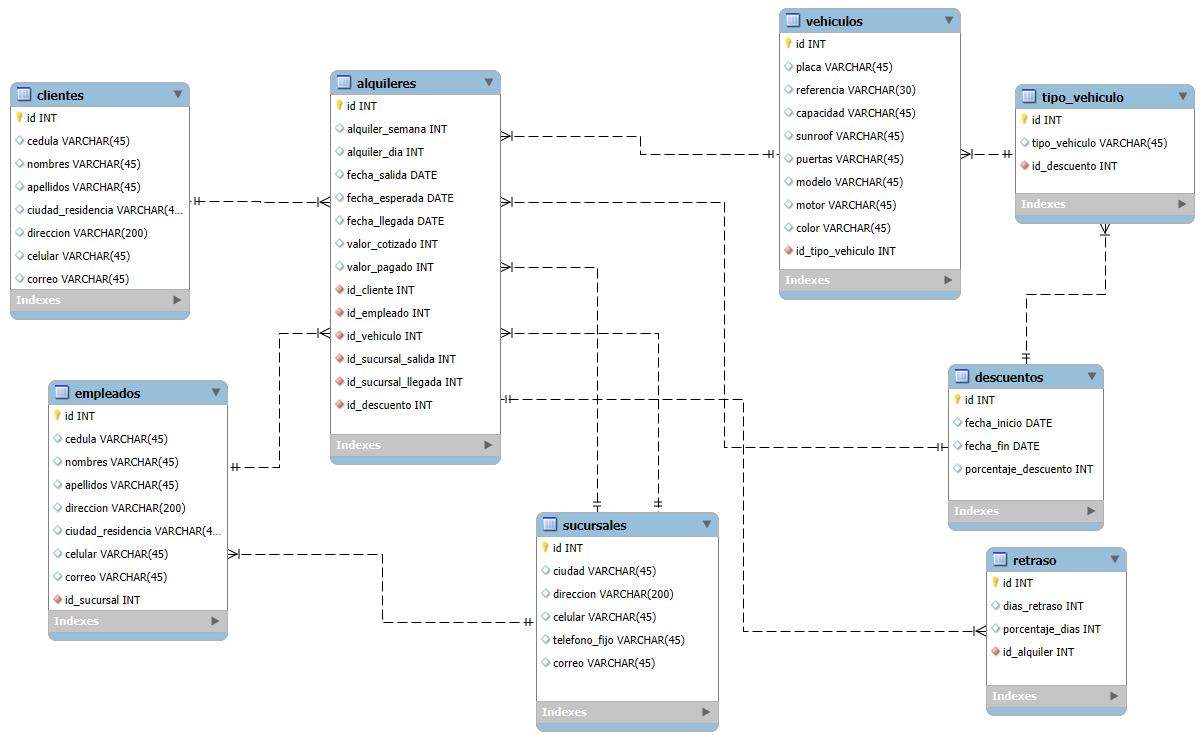
#### La segunda forma normal, es el segundo nivel de normalización en el diseño de la base de

#### 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.

No se necesito hacer mas tablas ya que las tablas estaban con sus atributos necesarios

#### Gráfica



#### 

## 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.

### Código

**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 clientes

create table clientes(

id int primary key,

cedula varchar(45) not null,

nombre1 varchar(30) not null,

nombre2 varchar(30) null,

apellido1 varchar(30) not null,

apellido2 varchar(30) null,

ciudad\_residencia varchar(45) not null,

direccion varchar(200) not null,

celular varchar(45) not null,

correo varchar(45) null);

2. Creación de la tabla sucursales

create table sucursales(

id int primary key,

ciudad varchar(45) not null,

direccion varchar(200) not null,

celular varchar(45) not null,

telefono\_fijo varchar(45) null,

correo varchar(45) null

);

3. Creación de la tabla empleados

create table empleados(

id int primary key,

cedula varchar(45) not null,

nombre1 varchar(30) not null,

nombre2 varchar(30) null,

apellido1 varchar(30) not null,

apellido2 varchar(30) null,

direccion varchar(200) not null,

ciudad\_residencia varchar(45) not null,

celular varchar(45) not null,

correo varchar(45) null,

id\_sucursal int,

foreign key (id\_sucursal) references sucursales (id));

4. Creación de la tabla descuentos

create table descuentos(

id int primary key,

fecha\_inicio date null,

fecha\_fin date null,

porcentaje\_descuento int null

);

5. Creación de la tipo\_vehiculo

create table tipo\_vehiculo(

id int primary key,

tipo\_vehiculo varchar(45) not null,

id\_descuento int,

foreign key (id\_descuento) references descuentos (id)

);

6. Creación de la tabla vehiculos

create table vehiculos(

id int primary key,

placa varchar(45) not null,

referencia varchar(45) not null,

capacidad varchar(45) not null,

sunroof varchar(45) not null,

puertas varchar(45) not null,

modelo varchar(45) not null,

motor varchar(45) not null,

color varchar(45) not null,

id\_tipo\_vehiculo int,

foreign key (id\_tipo\_vehiculo) references tipo\_vehiculo (id)

);

7. Creación de la tabla alquileres

create table alquileres(

id int primary key,

alquiler\_semana int not null,

alquiler\_dia int not null,

id\_sucursal\_salida int,

fecha\_salida date not null,

id\_sucursal\_llegada int,

fecha\_esperada date not null,

fecha\_llegada date null,

id\_cliente int,

id\_empleado int,

id\_vehiculo int,

valor\_cotizado int not null,

valor\_pagado int not null,

id\_descuento int,

foreign key (id\_sucursal\_salida) references sucursales (id),

foreign key (id\_sucursal\_llegada) references sucursales (id),

foreign key (id\_cliente) references clientes (id),

foreign key (id\_empleado) references empleados (id),

foreign key (id\_descuento) references descuentos (id)

);

8. Creación de la tabla retraso

create table retraso(

id int primary key,

dias\_retraso int null,

porcentaje\_dias int null,

id\_alquiler int,

foreign key (id\_alquiler) references alquileres (id));

Finalmente para listar las tablas anteriormente creadas utilizamos este comando:

show tables;

Construcción del Diagrama UML

Se ha diseñado un diagrama UML tomando como referencia la normalización para entender

mejor los diseños, la arquitectura del código y la implementación propuesta. Este enfoque nos

permitirá tener una visión clara y detallada de cómo se manejarán cada una de las consultas,

funcionalidades y los usuarios en la base de datos. De esta manera, podremos asegurar una

implementación eficiente y coherente con los requisitos del sistema.

Descripción

El diagrama UML se ha diseñado con el objetivo de representar detalladamente la estructura de

cada tabla y sus relaciones. Este diagrama ilustra claramente el tipo de dato correspondiente a

cada atributo, así como la identificación de claves primarias (primary keys) y claves foráneas

(foreign keys). Además, especifica la obligatoriedad de los atributos, proporcionando una visión

precisa y exhaustiva de la base de datos. Este nivel de detalle facilita el entendimiento y la

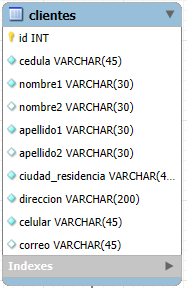
implementación técnica, asegurando que todos los componentes y sus interrelaciones estén

correctamente definidos y alineados con los requisitos del sistema.

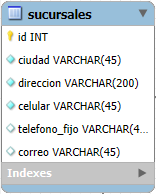
Comenzaremos creando las tablas junto con sus tipos de datos correspondientes:

#### Gráfica

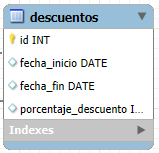
1. Tabla clientes



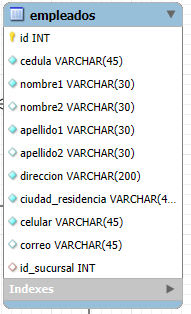
1. Tabla sucursales



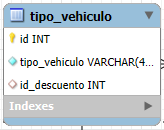
1. Tabla descuentos



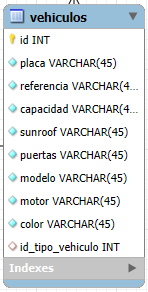
4. Tabla empleados



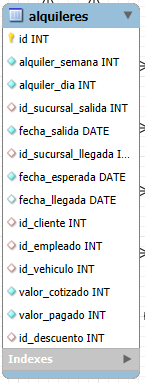
5.Tabla tipo\_vehiculo



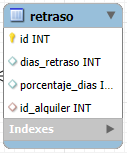
6. Tabla vehiculos

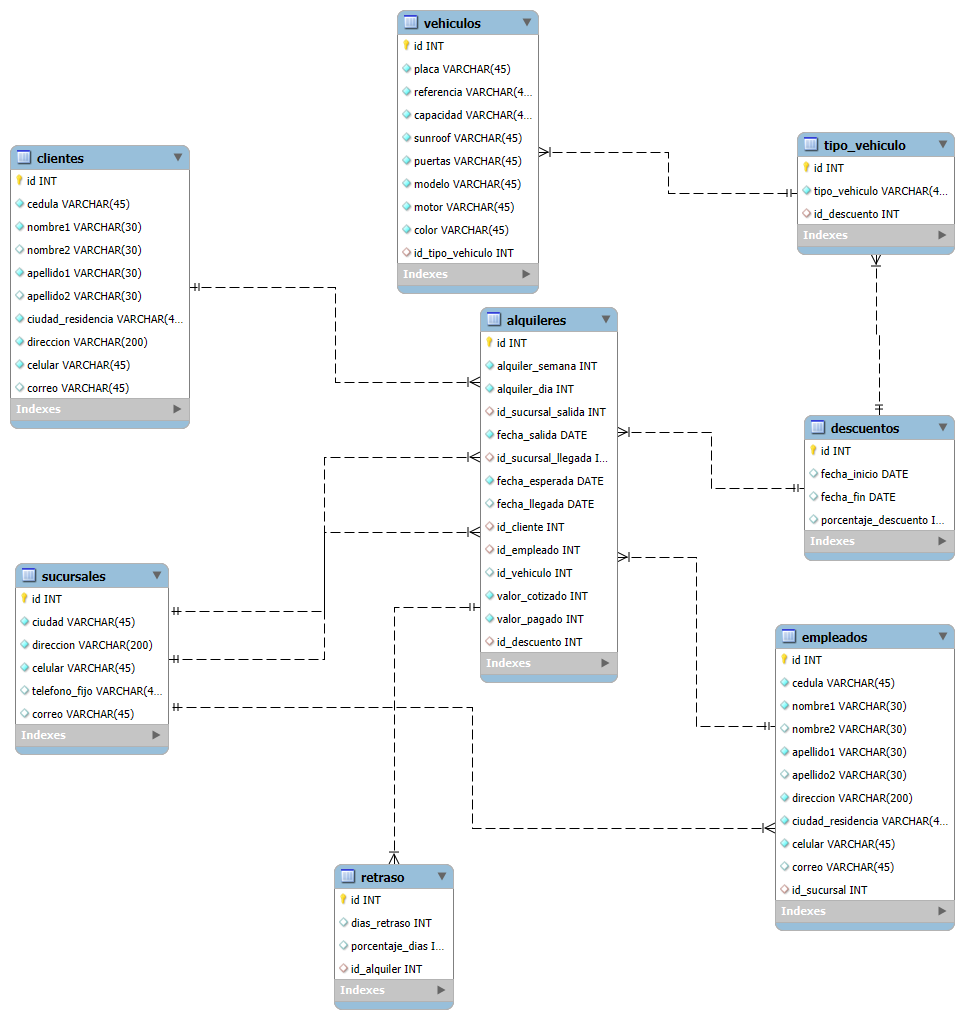


7. Tabla alquileres



8.Tabla retraso





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.

37

❖ 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

Para tener en cuenta se crearon unos datos ficticios en el contexto real para poder hacer la insercion de los mismos.

Tabla clientes

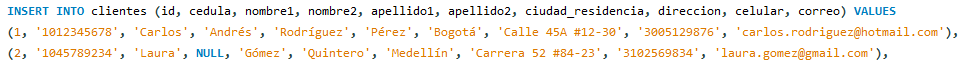


Tabla sucursales



Tabla empleados



Tabla descuentos



Tabla tipo\_vehiculo



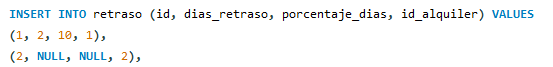
Tabla vehiculos



Tabla alquileres



Tabla retraso



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. Para realizar

consultas básicas,se utiliza la siguiente sintaxis:

En esta consulta se muestran todos los datos ‘SELECT \*’ de una tabla en especifico ‘FROM

nombre\_tabla’ :

SELECT \* FROM nombre\_tabla;

En esta consulta se muestran todos los datos ‘SELECT \*’ de una tabla en especifico ‘FROM

nombre\_tabla’ donde el valor sea igual al de la columna especificada WHERE columna =

'valor'.

SELECT \* FROM nombre\_tabla WHERE columna = 'valor';

###### Gráfica

# Referencias