

AUTORENTAL

WILMER ALEXANDER ROJAS FUENTES

T1

PEDRO FELIPE GÓMEZ BONILLA

**CAMPUSLANDS
SANDBOX
RUTA JAVA
TIBU
2024**

Tabla de Contenidos

Introducción	4
Caso de Estudio	5
Planificación	5
Construcción del Modelo Conceptual	5
Descripción	5
Gráfica	5
Descripción Técnica	6
Construcción del Modelo Lógico	6
Descripción	6
Gráfica	6
Descripción Técnica	6
Normalización del Modelo Lógico	6
Primera Forma Normal (1FN)	7
Descripción	7
Gráfica	7
Descripción Técnica	7
Segunda Forma Normal (2FN)	7
Descripción	8
Gráfica	8
Descripción Técnica	8
Tercera Forma Normal (3FN)	8
Descripción	8
Gráfica	8
Descripción Técnica	8
Construcción del Modelo Físico	9
Descripción	9
Código	9
Descripción Técnica	10
Diagrama E-R	10
Descripción	10
Gráfica	10
Descripción Técnica	10
Tablas	11
Descripción	11
Gráfica	11
Descripción Técnica	11
Relaciones entre Tablas	11
Descripción	11
Gráfica	12
Descripción Técnica	12
Inserción de Datos	12
Descripción	12

	Gráfica	12
	Descripción Técnica	12
Referencias		13

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:

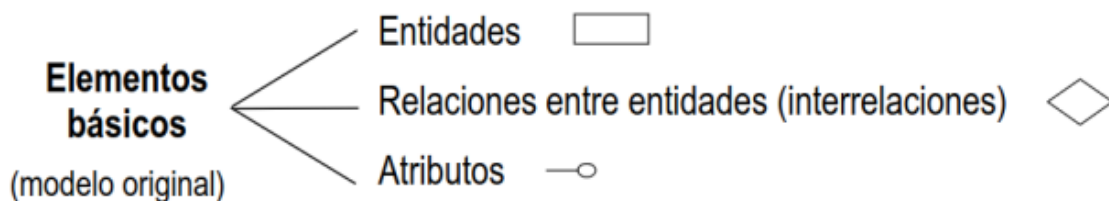
- Sucursales
- Empleados
- Clientes
- Vehículos
- Alquileres

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.

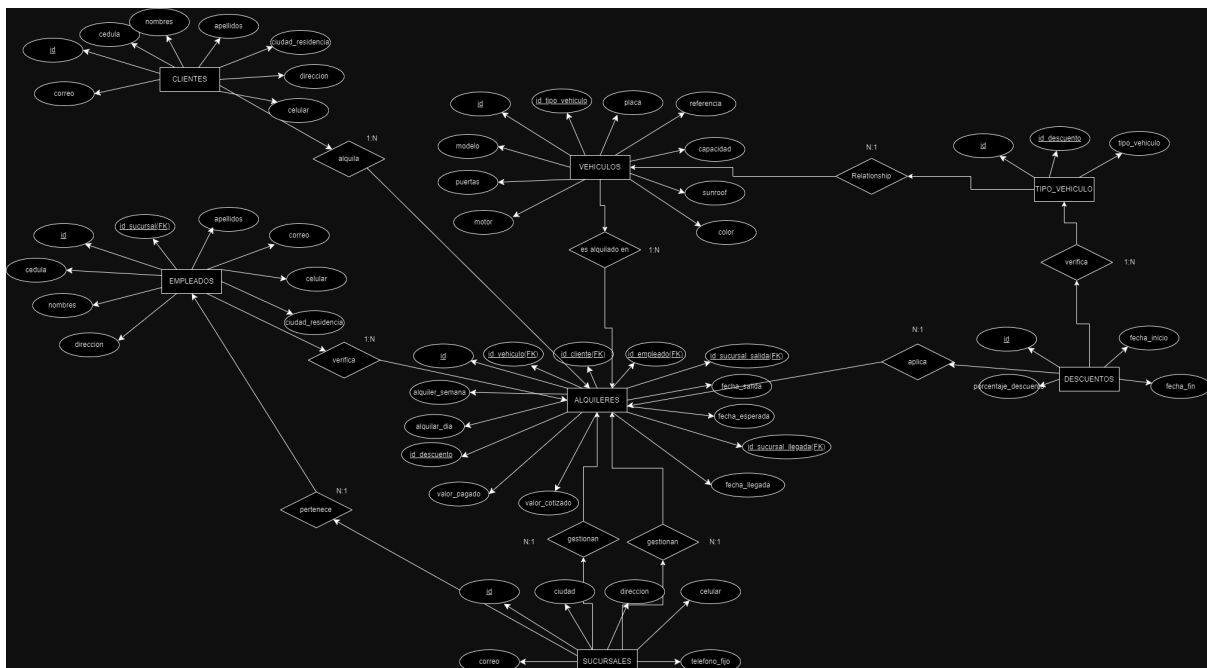
vehículos - alquileres

- **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).

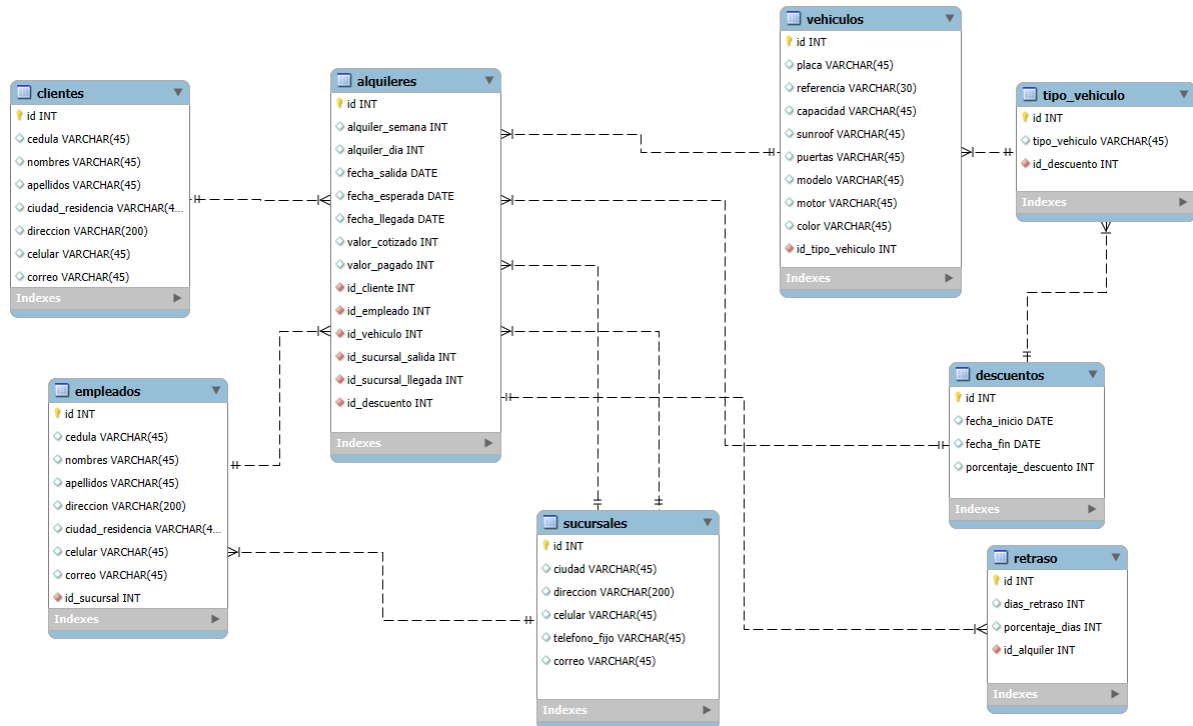
Alquileres:

- **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

			clientes								
id(PK)	cedula	nombre1	nombre2	apellido1	apellido2	ciudad_residencia	direccion	celular	correo		
INT	VARCHAR(45)	VARCHAR(30)	VARCHAR(30)	VARCHAR(30)	VARCHAR(30)	VARCHAR(45)	VARCHAR(200)	VARCHAR(45)	VARCHAR(45)	VARCHAR(45)	

			empleados								
id(PK)	cedula	nombre1	nombre2	apellido1	apellido2	direccion	ciudad_residencia	celular	correo	id_sucursal(FK)	
INT	VARCHAR(45)	VARCHAR(30)	VARCHAR(30)	VARCHAR(30)	VARCHAR(30)	VARCHAR(200)	VARCHAR(45)	VARCHAR(45)	VARCHAR(45)	INT	

alquileres														
id(PK)	alquiler_semana	alquiler_dia	fecha_salida	fecha_esperada	fecha_llegada	valor_cotizado	valor_pagado	id_cliente(FK)	id_empleado(FK)	id_vehiculo(FK)	id_sucursal_salida(FK)	id_sucursal_llegada(FK)	id_descuento(FK)	
INT	INT	INT	DATE	DATE	DATE	INT	INT	INT	INT	INT	INT	INT	INT	

		sucursales							
id(PK)	ciudad	direccion	celular	telefono_fijo	correo				
INT	VARCHAR(45)	VARCHAR(200)	VARCHAR(45)	VARCHAR(45)	VARCHAR(45)				

		vehiculos									
id(PK)	placa	referencia	capacidad	sunroof	puertas	modelo	motor	color	id_tipo_vehiculo(FK)		
INT	VARCHAR(45)	VARCHAR(30)	VARCHAR(45)	VARCHAR(45)	VARCHAR(45)	VARCHAR(45)	VARCHAR(45)	VARCHAR(45)	INT		

tipo_vehiculo			
id(PK)	tipo_vehiculo	id_descuento(FK)	
INT	VARCHAR(45)	INT	

descuentos			
id(PK)	fecha_inicio	fecha_fin	porcentaje_descuento
INT	DATE	DATE	INT

descuentos			
id(PK)	fecha_inicio	fecha_fin	porcentaje_descuento
INT	DATE	DATE	INT

retraso			
id(PK)	dias_retraso	porcentaje_dias	id_alquiler
INT	INT	INT	INT

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

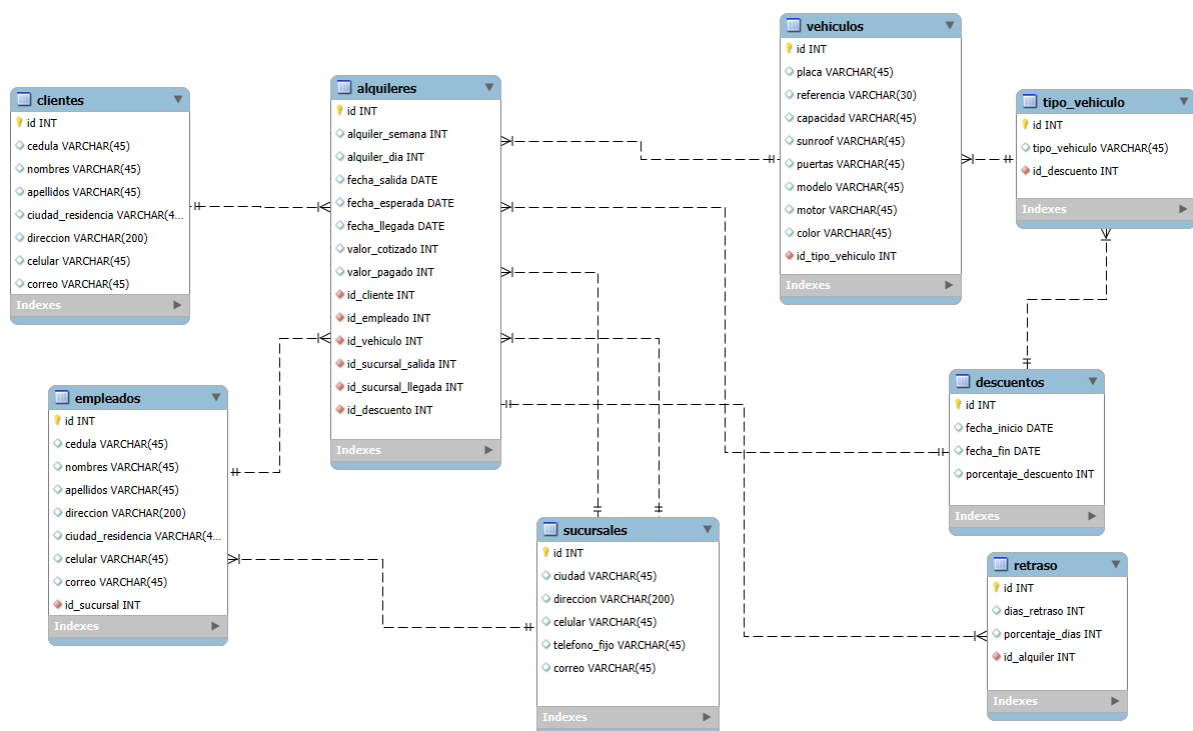
La segunda forma normal, es el segundo nivel de normalización en el diseño de la base de

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 necesita hacer más 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

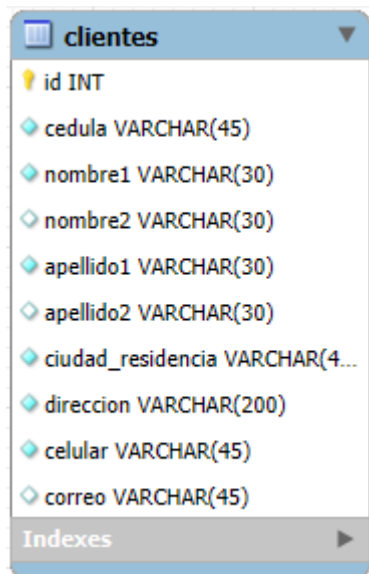
(foreign keys). Además, especifica la obligatoriedad de los atributos, proporcionando una visión

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

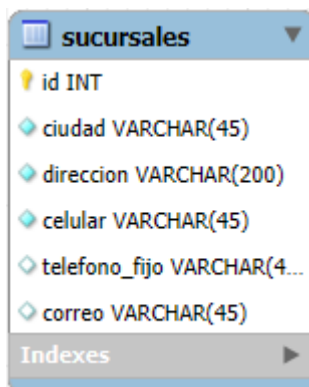


The screenshot shows the 'clientes' table with the following fields:

Field Name	Data Type
id	INT
cedula	VARCHAR(45)
nombre1	VARCHAR(30)
nombre2	VARCHAR(30)
apellido1	VARCHAR(30)
apellido2	VARCHAR(30)
ciudad_residencia	VARCHAR(45)
direccion	VARCHAR(200)
celular	VARCHAR(45)
correo	VARCHAR(45)

Below the fields, there is an 'Indexes' section with a right-pointing arrow.

2. Tabla sucursales

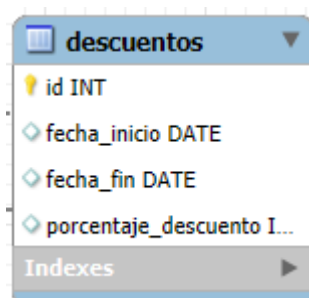


The screenshot shows the 'sucursales' table with the following fields:

Field Name	Data Type
id	INT
ciudad	VARCHAR(45)
direccion	VARCHAR(200)
celular	VARCHAR(45)
telefono_fijo	VARCHAR(45)
correo	VARCHAR(45)

Below the fields, there is an 'Indexes' section with a right-pointing arrow.

3. Tabla descuentos



The screenshot shows the 'descuentos' table with the following fields:

Field Name	Data Type
id	INT
fecha_inicio	DATE
fecha_fin	DATE
porcentaje_descuento	INT

Below the fields, there is an 'Indexes' section with a right-pointing arrow.

4. Tabla empleados

empleados	
id	INT
cedula	VARCHAR(45)
nombre1	VARCHAR(30)
nombre2	VARCHAR(30)
apellido1	VARCHAR(30)
apellido2	VARCHAR(30)
direccion	VARCHAR(200)
ciudad_residencia	VARCHAR(4...
celular	VARCHAR(45)
correo	VARCHAR(45)
id_sucursal	INT
Indexes	

5. Tabla tipo_vehiculo

tipo_vehiculo	
id	INT
tipo_vehiculo	VARCHAR(4...
id_descuento	INT
Indexes	

6. Tabla vehiculos

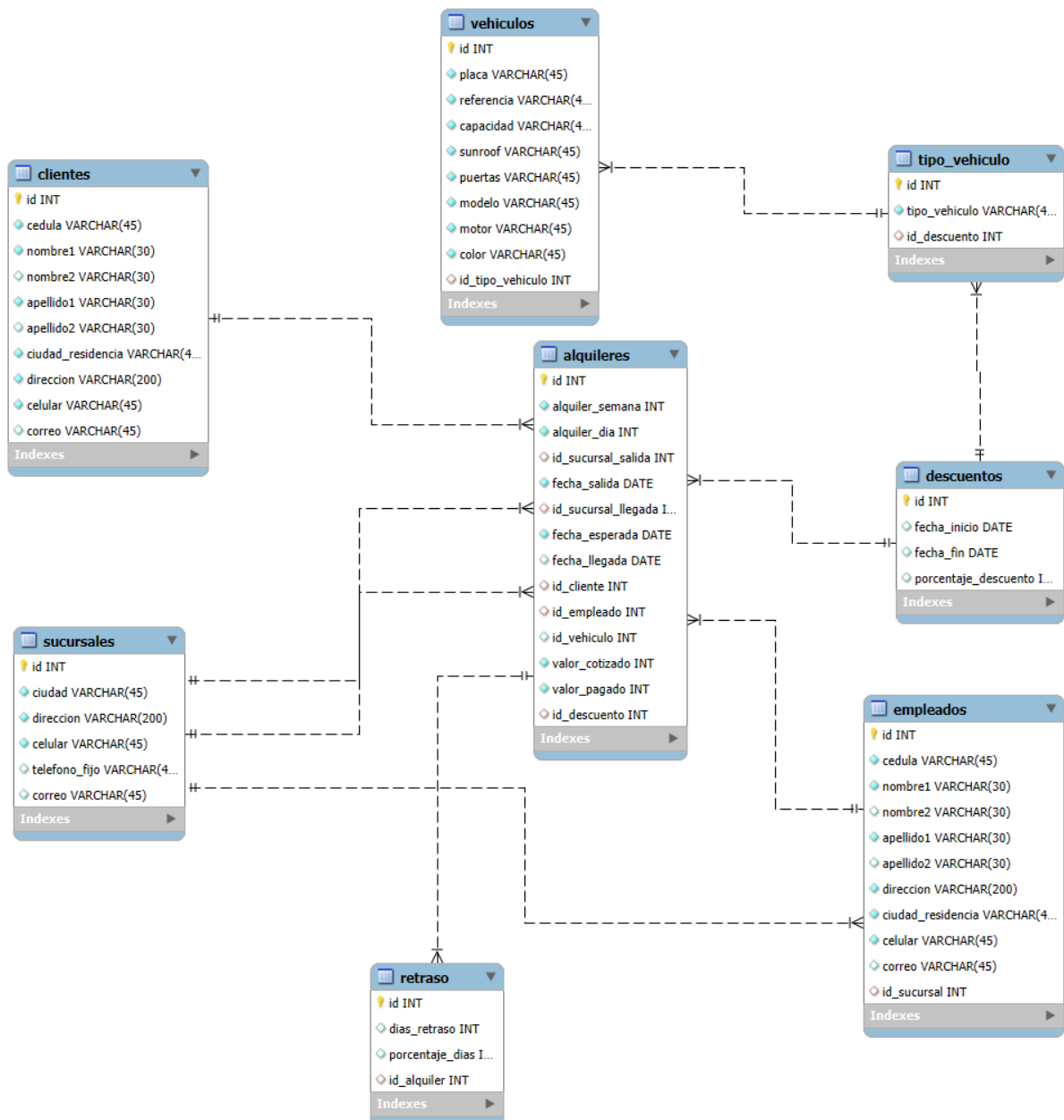
vehiculos	
id	INT
placa	VARCHAR(45)
referencia	VARCHAR(4...
capacidad	VARCHAR(4...
sunroof	VARCHAR(45)
puertas	VARCHAR(45)
modelo	VARCHAR(45)
motor	VARCHAR(45)
color	VARCHAR(45)
id_tipo_vehiculo	INT
Indexes	

7. Tabla alquileres

alquileres	
id	INT
alquiler_semana	INT
alquiler_dia	INT
id_sucursal_salida	INT
fecha_salida	DATE
id_sucursal_llegada	INT
fecha_esperada	DATE
fecha_llegada	DATE
id_cliente	INT
id_empleado	INT
id_vehiculo	INT
valor_cotizado	INT
valor_pagado	INT
id_descuento	INT
Indexes	

8.Tabla retraso

retraso	
id	INT
dias_retraso	INT
porcentaje_dias	INT
id_alquiler	INT
Indexes	



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

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

Tabla clientes

```
INSERT INTO clientes (id, cedula, nombre1, nombre2, apellido1, apellido2, ciudad_residencia, direccion, celular, correo) VALUES
(1, '1012345678', 'Carlos', 'Andrés', 'Rodríguez', 'Pérez', 'Bogotá', 'Calle 45A #12-30', '3005129876', 'carlos.rodriguez@hotmail.com'),
(2, '1045789234', 'Laura', NULL, 'Gómez', 'Quintero', 'Medellín', 'Carrera 52 #84-23', '3102569834', 'laura.gomez@gmail.com'),
```

Tabla sucursales

```
INSERT INTO sucursales (id, ciudad, direccion, celular, telefono_fijo, correo) VALUES
(1, 'Bogotá', 'Calle 45 #23-12', '3201234567', '6012345678', 'bogota@sucursales.co'),
(2, 'Medellín', 'Carrera 65 #32-45', '3159876543', '6041234567', 'medellin@sucursales.co'),
```

Tabla empleados

```
INSERT INTO empleados (id, cedula, nombre1, nombre2, apellido1, apellido2, direccion, ciudad_residencia, celular, correo, id_sucursal) VALUES
(1, '1012345678', 'Juan', 'Carlos', 'Gómez', 'Pérez', 'Calle 45 #12-34', 'Bogotá', '3201234567', 'juan.gomez@empresa.co', 1),
(2, '1023456789', 'María', 'Fernanda', 'Rodríguez', 'López', 'Carrera 65 #32-12', 'Medellín', '3152345678', 'maria.rodriguez@empresa.co', 2),
```

Tabla descuentos

```
INSERT INTO descuentos (id, fecha_inicio, fecha_fin, porcentaje_descuento) VALUES
(1, '2024-01-01', '2024-03-31', 10),
(2, '2024-04-01', '2024-06-30', 15),
```

Tabla tipo_vehiculo

```
INSERT INTO tipo_vehiculo (id, tipo_vehiculo, id_descuento) VALUES
(1, 'Sedán', 1),          -- Descuento 20%
(2, 'SUV', 2),           -- Descuento 15%
```

Tabla vehiculos

```
INSERT INTO vehiculos (id, placa, referencia, capacidad, sunroof, puertas, modelo, motor, color, id_tipo_vehiculo) VALUES
(1, 'ABC123', 'Sedán 2023', '5', 'Sí', '4', 'Sedán Turbo', '2.0', 'Negro', 1),
(2, 'DEF456', 'SUV 2022', '7', 'Sí', '5', 'SUV Familiar', '3.0', 'Blanco', 2),
```

Tabla alquileres

```
INSERT INTO alquileres (id, alquiler_semana, alquiler_dia, id_sucursal_salida, fecha_salida, id_sucursal_llegada, fecha_esperada, fecha_llegada, id_cliente, id_empleado, id_vehiculo, valor_cotizado, valor_pagado, id_descuento) VALUES
(1, 700000, 120000, 1, '2024-01-15', 3, '2024-01-22', '2024-01-21', 1, 1, 1, 840000, 820000, 1),
(2, 650000, 115000, 2, '2024-01-18', 4, '2024-01-25', NULL, 2, 2, 2, 700000, 750000, 2),
```

Tabla retraso

```
INSERT INTO retraso (id, dias_retraso, porcentaje_dias, id_alquiler) VALUES
(1, 2, 10, 1),
(2, NULL, NULL, 2),
```

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 específico 'FROM nombre_tabla' :

```
SELECT * FROM nombre_tabla;
```

En esta consulta se muestran todos los datos 'SELECT *' de una tabla en específico '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