**AUTORENTAL**

**MARIA GUADALUPE LIZARAZO LEAL**

**T2**

**PEDRO FELIPE GÓMEZ BONILLA**

**CAMPUSLANDS**

**SANDBOX - RUTA JAVA**

**TIBU**

**2024**

**Tabla de contenido**

[INTRODUCCIÓN 3](#_Toc182441756)

[Caso de Estudio 4](#_Toc182441757)

[Planificación 5](#_Toc182441758)

[Construcción del Modelo Conceptual 5](#_Toc182441759)

[Descripción 5](#_Toc182441760)

[Gráfica 8](#_Toc182441761)

[Construcción del Modelo Lógico 8](#_Toc182441762)

[Descripción 8](#_Toc182441763)

[Gráfica 11](#_Toc182441764)

[Normalización del Modelo Lógico 11](#_Toc182441765)

[Primera Forma Normal (1FN) 11](#_Toc182441766)

[Segunda Forma Normal (2FN) 13](#_Toc182441767)

[Tercer Forma Normal (3FN) 14](#_Toc182441768)

[Construcción del Modelo Físico 16](#_Toc182441769)

[Descripción 16](#_Toc182441770)

[Creación de las tablas: 16](#_Toc182441771)

[Construcción del Diagrama UML 19](#_Toc182441772)

[Descripción 19](#_Toc182441773)

[Tablas - Diagrama UML: 19](#_Toc182441774)

[Gráfica 22](#_Toc182441775)

# 

# INTRODUCCIÓN

La empresa AutoRental, dedicada al alquiler de vehículos, enfrenta el desafío de optimizar sus procesos de negocio para mejorar la eficiencia y experiencia del cliente. Con cinco sucursales en distintas ciudades y planes de expansión a nivel nacional; AutoRental cuenta con una flota diversificada que incluye vehículos de varios tipos, modelos y capacidades, adaptados a distintas necesidades de sus clientes. En este contexto, surge la necesidad de un sistema de información centralizado que permita gestionar eficazmente el inventario de vehículos, los alquileres y las políticas de precios y descuentos.

Este proyecto requiere una estructura estable y flexible que permita manejar características propias de AutoRental, como la posibilidad de alquilar un vehículo en una sucursal y devolverlo en otra, aplicar descuentos específicos para ciertos vehículos durante el año y gestionar tarifas de alquiler basadas en el tipo de vehículo, con cobros adicionales por retraso en la entrega.

Seguido de esto, procederemos a realizar el modelo conceptual y lógico; las formas normales, de las cuales especificaremos la 1FN, 2FN y 3FN. Finalizaremos creando la relación y cardinalidades; el código físico en MySQL y la construcción del modelo UML.

# Caso de Estudio

La empresa AutoRental, nos ha planteado unas especificaciones de las cuales nos basaremos, analizando y poniendo énfasis en todos los requerimientos y solicitudes.

Inicialmente, empezaremos con el análisis de la empresa, teniendo en cuenta los siguientes requerimientos:

* **Sucursales:** ciudad y dirección donde se ubica, teléfono fijo, celular y correo electrónico.
* **Empleados:** sucursal donde labora, cédula, nombres, apellidos, dirección y ciudad de residencia, celular y correo electrónico.
* **Clientes:** cédula, nombres, apellidos, dirección y ciudad de residencia, celular y correo electrónico.
* **Vehículos**: tipo de vehículo, placa, referencia, modelo, puertas, capacidad, sunroof, motor, color.
* **Alquileres:** vehículo, cliente, empleado, sucursal y fecha de salida, sucursal y fecha de llegada, fecha esperada de llegada, valor de alquiler por semana, valor de alquiler por día, porcentaje de descuento, valor cotizado y valor pagado.

# Planificación

## Construcción del Modelo Conceptual

Para construir el modelo conceptual tomaremos como base los atributos proporcionados en el punto de partida. Tendremos en cuenta las posibles relaciones y aplicaremos los requerimientos propuestos.

Así mismo, para dar más claridad de lo que es un modelo conceptual, primero debemos entender los elementos básicos que se utilizarán para desarrollarlo:



Elementos Básicos

(Modelo Original)





### Descripción

El modelo conceptual propuesto está basado en los diferentes atributos iniciales propuestos, se tiene en cuenta el funcionamiento básico y las relaciones.

Entidades y atributos:

1. **Sucursales**:

* IdSucursales: identificación única de cada sucursal.
* Ciudad: Ciudad en la que se encuentra dicha sucursal.
* Dirección: Direccion donde se encuentra la sucursal.
* Telefono\_Fijo: Número fijo de contacto de la sucursal.
* Celular: Número celular de contacto de la sucursal.
* Email: correo electrónico de contacto con la sucursal.

1. **Empleados**:

* IdEmpleados: identificación única de cada empleado.
* Nombre: nombres del empleado.
* Apellido: apellidos del empleado.
* Cedula: N° de Identificación del empleado.
* Celular: número de contacto del empleado.
* Dirección: dirección de residencia del empleado.
* Ciudad\_Residencia: Lugar donde vive el empleado.
* Email: correo electrónico de contacto con el empleado.
* idSucursales: Foreign Key tabla Sucursales.

1. **Clientes**:

* IdClientes: identificación única por cliente.
* Nombre: nombres del cliente.
* Apellido: apellidos del cliente.
* Cedula: N° de Identificación del cliente.
* Celular: número de contacto del cliente.
* Dirección: dirección de residencia del cliente.
* Ciudad\_Residencia: Lugar donde vive el cliente.
* Email: correo electrónico de contacto con el cliente.
* idEmpleados: Foreign Key tabla Empleados.

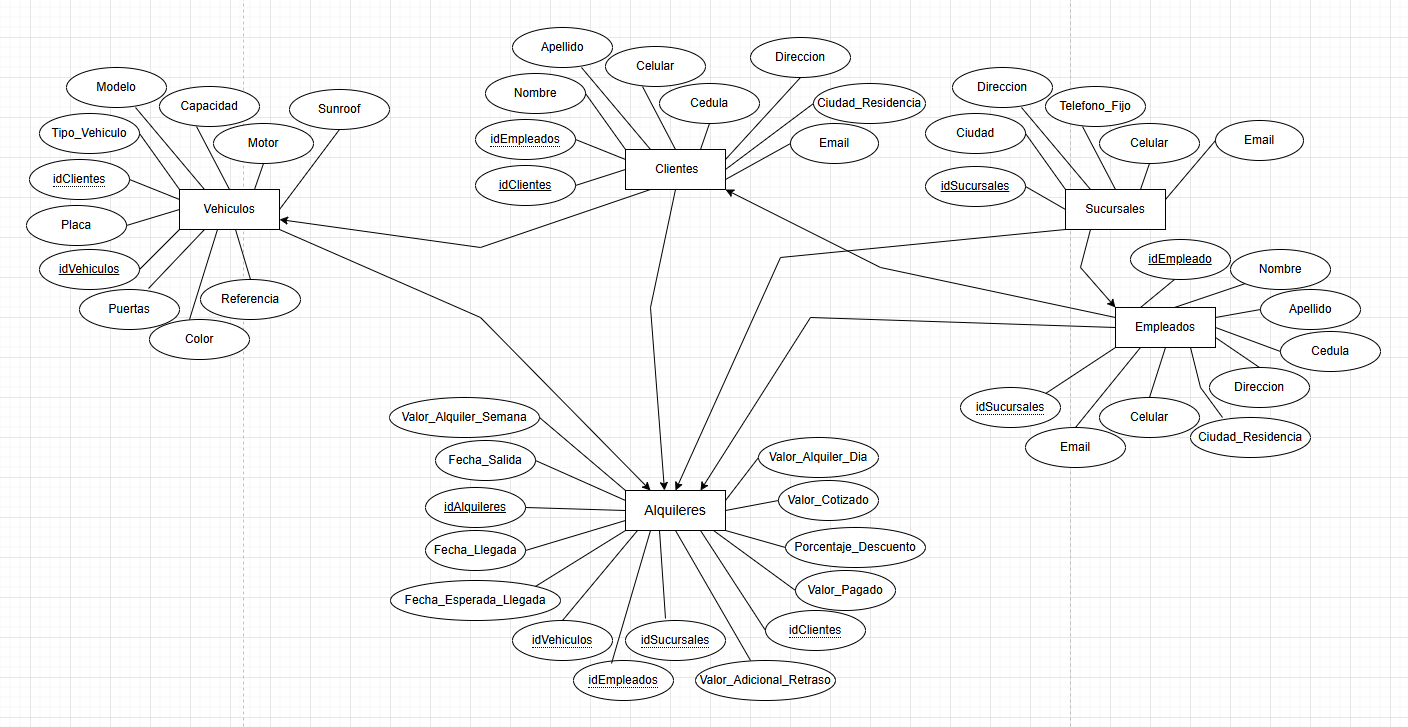
1. **Vehículos**:

* IdVehiculos: identificación única de cada vehículo.
* Tipo\_Vehiculo: Clasificación del vehículo (Sedán, Compacto, Camioneta Platón, Camioneta de Lujo, Deportivo, etc.).
* Modelo: Año de fabricación del vehículo.
* Capacidad: Número de pasajeros que puede transportar el vehículo.
* Placa: Número único de placa del vehículo
* Motor: Tipo y potencia del motor
* Sunroof: Indica si el vehículo cuenta con un techo solar o panorámico.
* Puertas: Número de puertas del vehículo.
* Color: Color del vehículo
* Referencia: Descripción específica del vehículo.
* idClientes: Foreign Key tabla Clientes.

1. **Alquileres**:

* IdAlquileres: identificación única de cada alquiler.
* Fecha\_Salida: Fecha en la que el vehículo es retirado por el cliente en la sucursal donde se encuentra.
* Fecha\_Llegada: Fecha real de devolución del vehículo en la sucursal de destino.
* Fecha\_Esperada\_Llegada: Fecha programada para la devolución del vehículo según el acuerdo de alquiler.
* Valor\_Alquiler\_Semana: Costo del alquiler del vehículo por semana.
* Valor\_Alquiler\_Dia: Costo del alquiler del vehículo por día.
* Porcentaje\_Descuento: Porcentaje de descuento aplicable al alquiler.
* Valor\_Cotizado: Monto total calculado inicialmente para el período de alquiler solicitado.
* Valor\_Pagado: Monto que el cliente ha pagado efectivamente por el alquiler del vehículo.
* Valor\_Adicional\_Retraso: Cargo adicional generado si el cliente devuelve el vehículo después de la Fecha\_Esperada\_Llegada.
* idVehiculos: Foreign Key tabla Vehículos.
* idEmpleados: Foreign Key tabla Empleados.
* idSucursales: Foreign Key tabla Sucursales.
* idClientes: Foreign Key tabla Clientes.

### Gráfica



## Construcción del Modelo Lógico

Para la creación del modelo lógico nos basamos en el esqueleto del modelo conceptual, aplicaremos los atributos previstos y empezaremos a formar las bases de la base de datos, incluyendo la entidad relación que habrá entre ellas, añadiendo y mostrando las llaves (Primary Key, Foreign Key).

### Descripción

El modelo lógico se realizará teniendo como base el modelo conceptual previamente realizado, con las llaves, tanto primaria cómo foránea; así mismo, las entidades relación de cardinalidad.

1. **Sucursales**:

* IdSucursales: INT PRIMARY KEY
* Ciudad: VARCHAR (45) NOT NULL
* Dirección: VARCHAR (45) NOT NULL
* Telefono\_Fijo: VARCHAR (45) NOT NULL
* Celular: INT NOT NULL
* Email: VARCHAR (45) NOT NULL

1. **Empleados**:

* IdEmpleados: INT PRIMARY KEY
* Nombre: VARCHAR (45) NOT NULL
* Apellido: VARCHAR (45) NOT NULL
* Cedula: INT NOT NULL
* Celular: INT NOT NULL
* Dirección: VARCHAR (45) NOT NULL
* Ciudad\_Residencia: VARCHAR (45) NOT NULL
* Email: VARCHAR (45) NOT NULL
* idSucursales: INT FOREIGN KEY NOT NULL.

1. **Clientes**:

* IdClientes: INT PRIMARY KEY
* Nombre: VARCHAR (45) NOT NULL
* Apellido: VARCHAR (45) NOT NULL
* Cedula: INT NOT NULL
* Celular: INT NOT NULL
* Dirección: VARCHAR (45) NOT NULL.
* Ciudad\_Residencia: VARCHAR (45) NOT NULL.
* Email: VARCHAR (45) NOT NULL
* idEmpleados: INT FOREIGN KEY NOT NULL

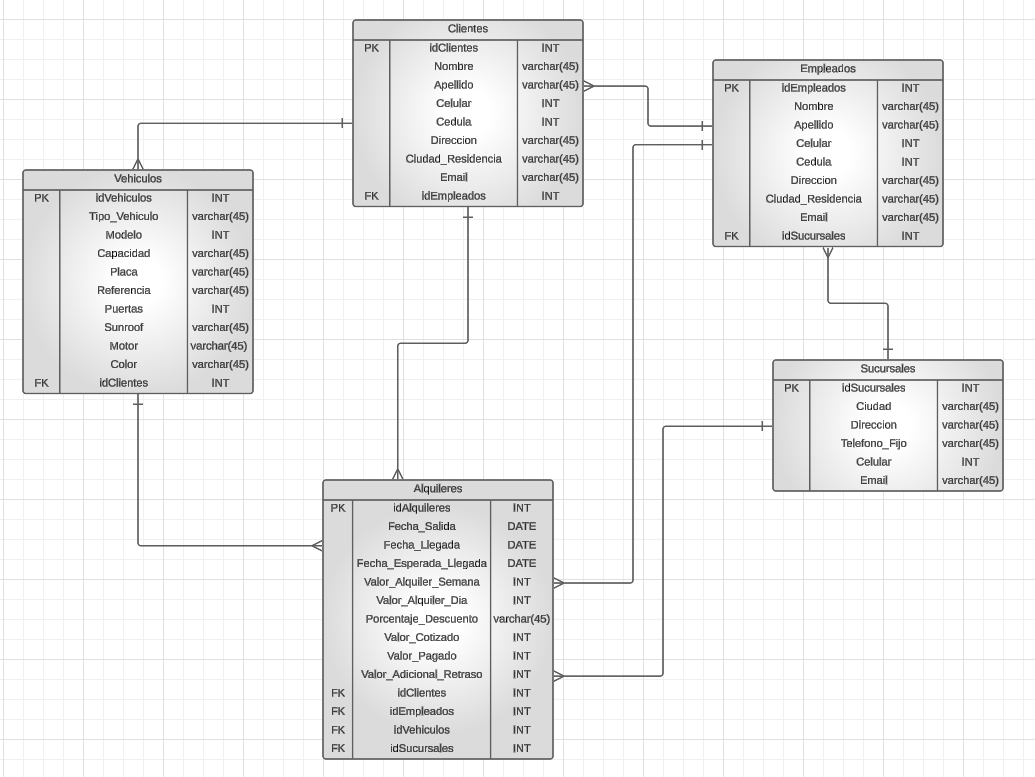
1. **Vehículos**:

* IdVehiculos: INT PRIMARY KEY
* Tipo\_Vehiculo: VARCHAR (45) NOT NULL
* Modelo: INT NOT NULL
* Capacidad: VARCHAR (45) NOT NULL
* Motor: VARCHAR (45) NOT NULL
* Sunroof: VARCHAR (45) NOT NULL
* Puertas: INT NOT NULL
* Color: VARCHAR (45) NOT NULL
* Referencia: VARCHAR (45) NOT NULL
* idClientes: INT FOREIGN KEY NOT NULL

1. **Alquileres**:

* IdAlquileres: INT PRIMARY KEY
* Fecha\_Salida: DATE
* Fecha\_Llegada: DATE
* Fecha\_Esperada\_Llegada: DATE
* Valor\_Alquiler\_Semana: INT NOT NULL
* Valor\_Alquiler\_Dia: INT NOT NULL
* Porcentaje\_Descuento: VARCHAR (45) NOT NULL
* Valor\_Cotizado: INT NOT NULL
* Valor\_Pagado: INT NOT NULL
* Valor\_Adicional\_Retraso: INT NOT NULL
* idVehiculos: INT FOREIGN KEY NOT NULL
* idEmpleados: INT FOREIGN KEY NOT NULL
* idSucursales: INT FOREIGN KEY NOT NULL
* idClientes: INT FOREIGN KEY NOT NULL

### Gráfica



## Normalización del Modelo Lógico

En esta parte pasamos a crear las tablas de normalización; de esta manera llevándose a cabo las 3 formas normales y así mismo, evitando de la mejor manera las redundancias y dependencias transitivas en nuestra base de datos.

### Primera Forma Normal (1FN)

Tener en cuenta que la 1FN se cumple si:

1. Sus atributos son atómicos.
2. No hay grupos repetitivos de columnas.
3. Cada columna obtiene un solo valor en cada fila.

#### Descripción

En la 1FN empezamos viendo si hay algún dato que se repita o no sea atómico, para así poder separarlos y dirigirnos directamente a la 2FN.

#### Descripción Técnica

1. **Clientes**:

* Se encuentra en la 1FN, debido a que ya se le ha puesto la llave primaria, cada columna ya ha sido previamente analizada y no tiene datos repetitivos.

1. **Vehículos**:

* Se encuentra en la 1FN, debido a que ya se le ha puesto la llave primaria, cada columna ya ha sido previamente analizada y no tiene datos repetitivos.

1. **Sucursales**:

* Se encuentra en la 1FN, debido a que ya se le ha puesto la llave primaria, cada columna ya ha sido previamente analizada y no tiene datos repetitivos.

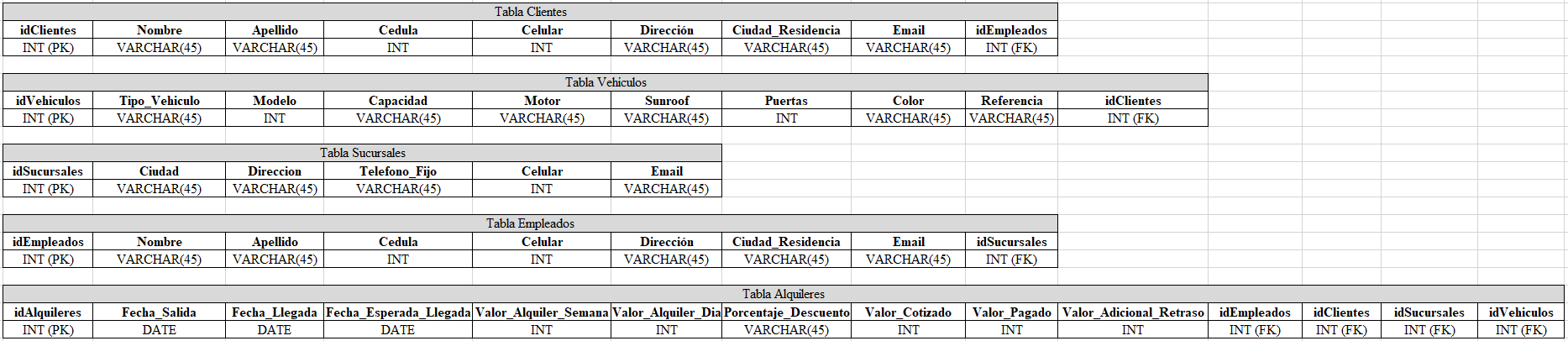
1. **Empleados**:

* Se encuentra en la 1FN, debido a que ya se le ha puesto la llave primaria, cada columna ya ha sido previamente analizada y no tiene datos repetitivos.

1. **Alquileres**:

* Se encuentra en la 1FN, debido a que ya se le ha puesto la llave primaria, cada columna ya ha sido previamente analizada y no tiene datos repetitivos.

#### Gráfica



### Segunda Forma Normal (2FN)

Tener en cuenta que la 2FN se cumple, sí:

* Ya se encuentra en su 1FN.
* Cada atributo que no sea clave primaria deberá depender funcionalmente de ésta.

#### Descripción

La 2FN es el segundo paso a seguir dentro de la normalización lógica, la cual nos permitirá observar si hay algún dato independiente a la llave primaria. Una vez realizado esto, nos dirigiremos a la última forma normal.

#### Descripción Técnica

1. **Clientes**:

* Se encuentra en la 2FN, debido a que cuenta con una llave primaria única a la que todos los atributos son dependientes.

1. **Vehículos**:

* Se encuentra en la 2FN, debido a que cuenta con una llave primaria única a la que todos los atributos son dependientes.

1. **Sucursales**:

* Se encuentra en la 2FN, debido a que cuenta con una llave primaria única a la que todos los atributos son dependientes.

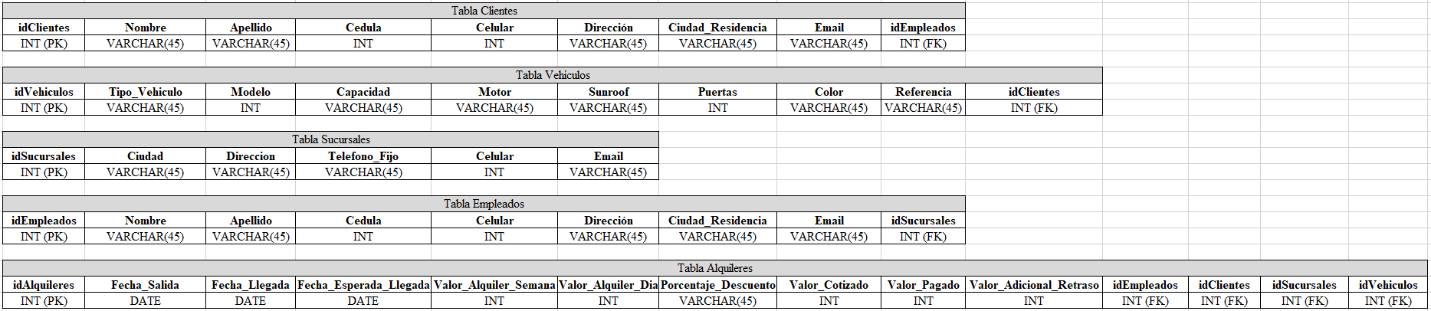
1. **Empleados**:

* Se encuentra en la 2FN, debido a que cuenta con una llave primaria única a la que todos los atributos son dependientes.

1. **Alquileres**:

* Se encuentra en la 2FN, debido a que cuenta con una llave primaria única a la que todos los atributos son dependientes.

#### Gráfica



### Tercer Forma Normal (3FN)

La 3FN se cumple si:

* Está en 2FN.
* Que ningún atributo no clave, dependa de otro no clave.

#### Descripción

La 3FN, es el último nivel de normalización que trabajaremos en el diseño de la base de datos que se aplicará a las tablas que ya cumplen con la 2FN y se enfoca en la eliminación de dependencias redundantes, evitando que un atributo no clave dependa de otro no clave.

#### Descripción Técnica

1. **Clientes**:

* Se encuentra en la 3FN, ya que previamente se hizo la 2FN y en cada columna no hay ningún atributo no clave que dependa de otro no clave.

1. **Vehículos**:

* Se encuentra en la 3FN, ya que previamente se hizo la 2FN y en cada columna no hay ningún atributo no clave que dependa de otro no clave.

1. **Sucursales**:

* Se encuentra en la 3FN, ya que previamente se hizo la 2FN y en cada columna no hay ningún atributo no clave que dependa de otro no clave.

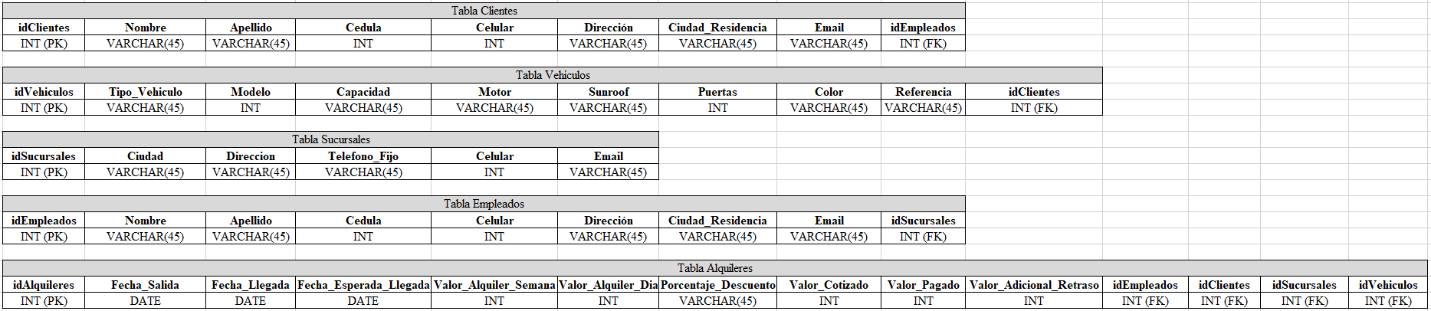
1. **Empleados**:

* Se encuentra en la 3FN, ya que previamente se hizo la 2FN y en cada columna no hay ningún atributo no clave que dependa de otro no clave.

1. **Alquileres**:

* Se encuentra en la 3FN, ya que previamente se hizo la 2FN y en cada columna no hay ningún atributo no clave que dependa de otro no clave.

#### Gráfica



## Construcción del Modelo Físico

Se diseñará el modelo físico teniendo en cuenta todos los cambios realizados en la normalización lógica. Para este diseño, usamos la herramienta MySQL.

### Descripción

El modelo físico se diseñará para que su funcionamiento sea solo en MySQL, dónde se mostrará la creación de cada tabla, consiguiendo un resultado completo de todo el modelo.

### Creación de las tablas:

* Para la creación del DATABASE utilice el siguiente comando:



* Para poder trabajar sobre la BBDD anteriormente creada, utilizo el siguiente comando:

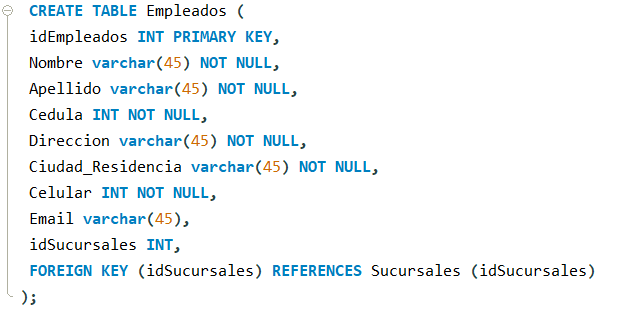


Después de la creación de la DATABASE, procedo a crear las tablas junto con sus atributos correspondientes. Para lo anterior mencionado, utilizo los siguientes comandos:

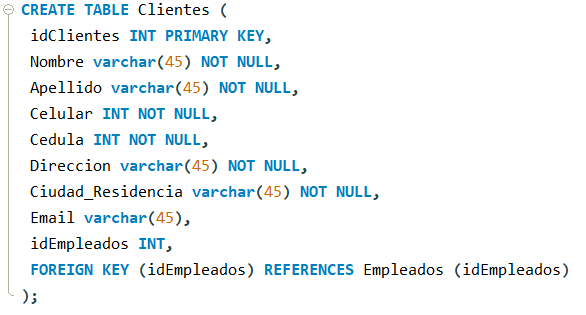
1. Creación de la tabla Sucursales:



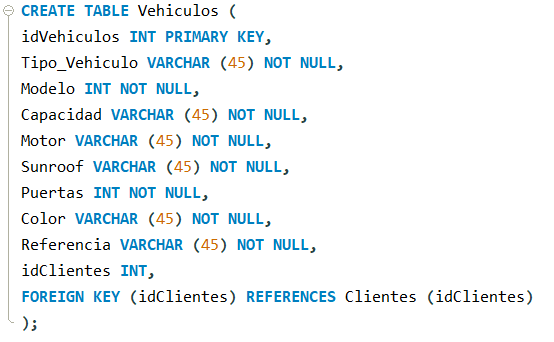
1. Creación de la tabla Empleados:



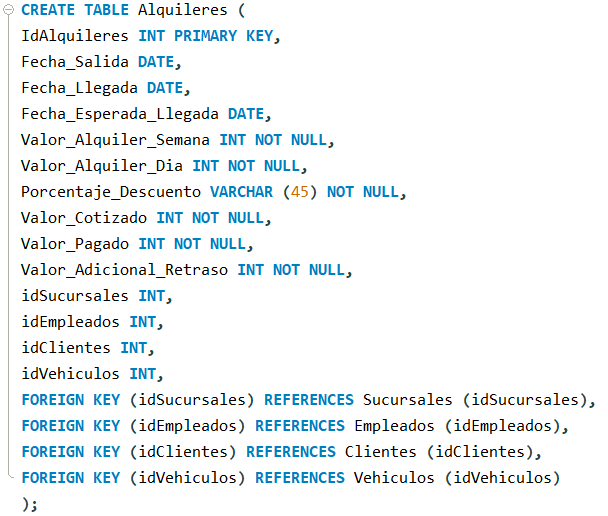
1. Creación de la tabla Clientes:



1. Creación de la tabla Vehículos:



1. Creación de la tabla Alquileres:



## Construcción del Diagrama UML

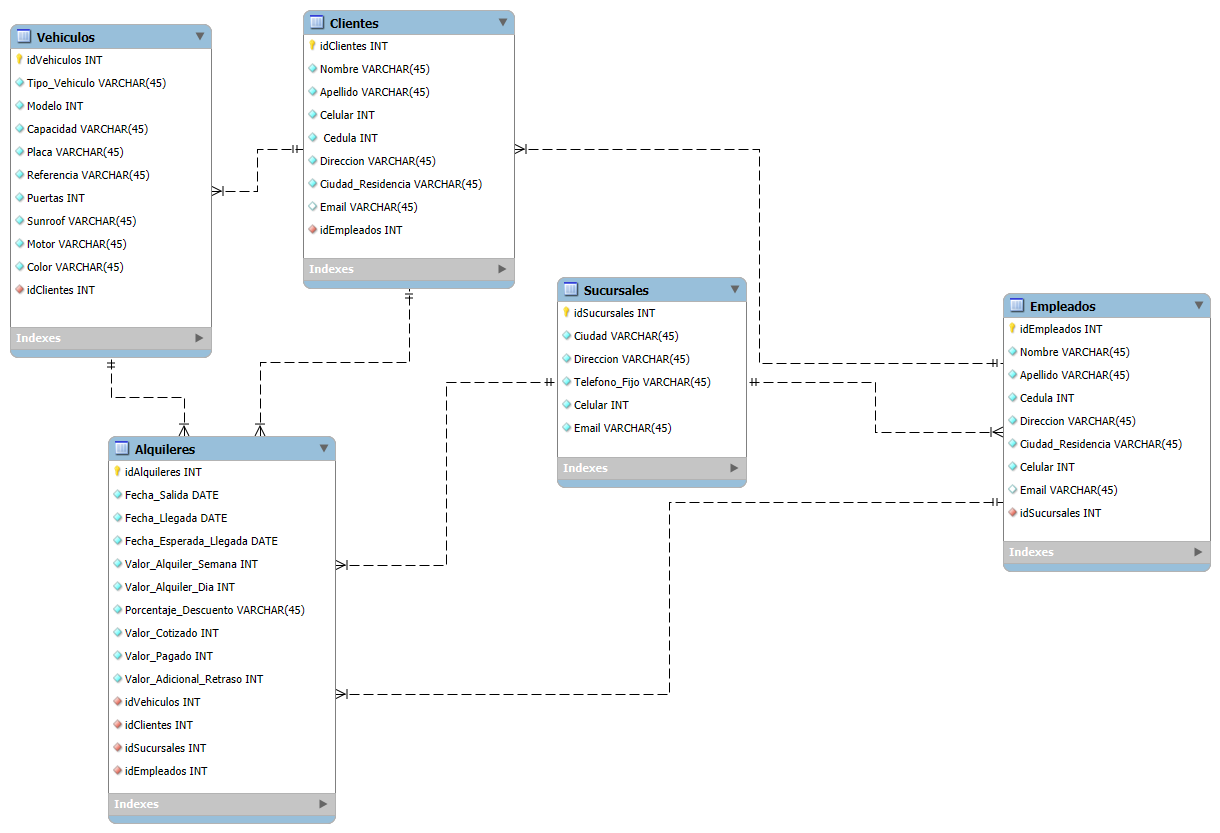
Se ha elaborado un diagrama UML, siguiendo los pasos ya realizados en la normalización, para facilitar la comprensión del diseño, la estructura del código y la implementación de BBDD propuesta. Este enfoque proporciona una visión directa de la gestión de consultas, funcionalidades y usuarios en la base de datos, garantizando así un desarrollo eficiente y alineado con los requisitos del sistema.

### Descripción

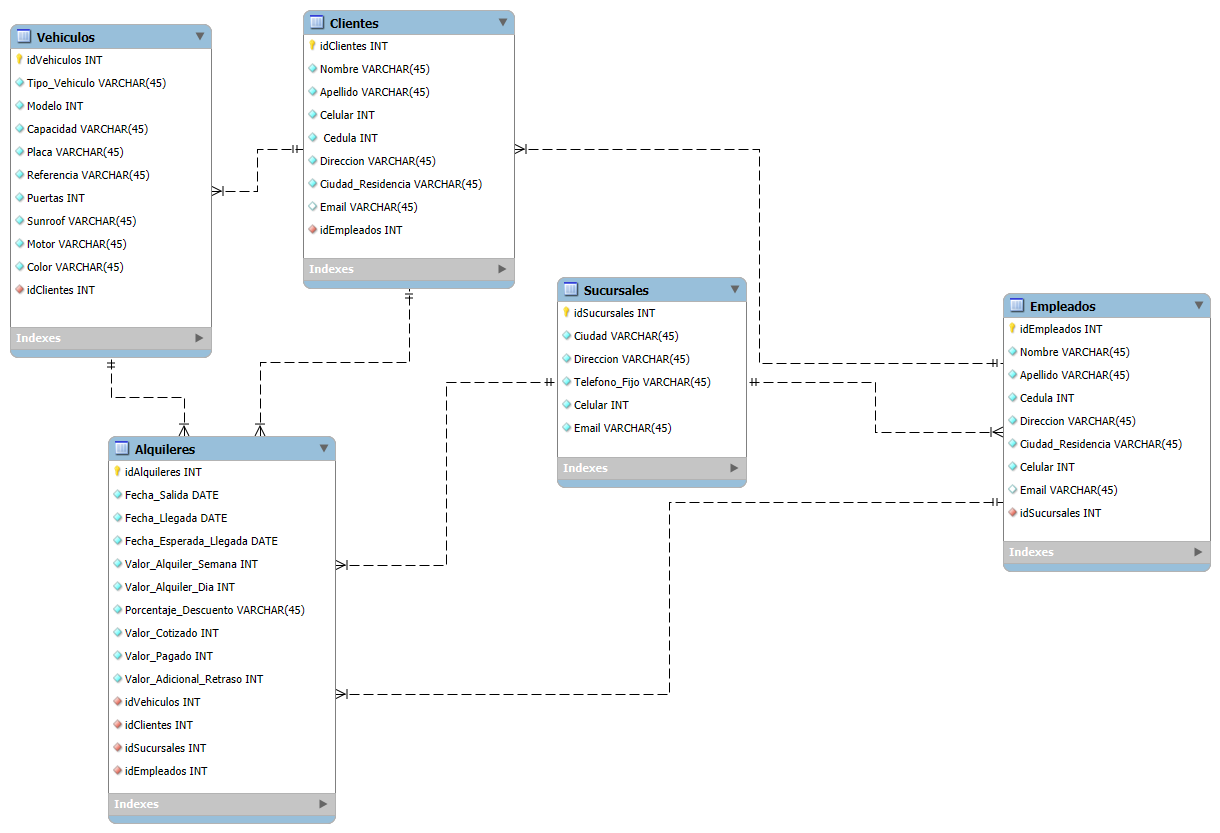
El diagrama UML se ha diseñado con el propósito de mostrar de manera detallada la estructura de cada tabla y sus relaciones. Este diagrama presenta claramente el tipo de dato asignado a cada atributo, además de identificar las claves primarias y foráneas. También indica la obligatoriedad de los atributos, ofreciendo una visión completa y precisa de la base de datos. Este nivel de desglose, facilita tanto la comprensión como la implementación técnica, asegurando que todos los elementos y sus interrelaciones estén bien definidos de acuerdo con los requisitos del sistema.

### Tablas - Diagrama UML:

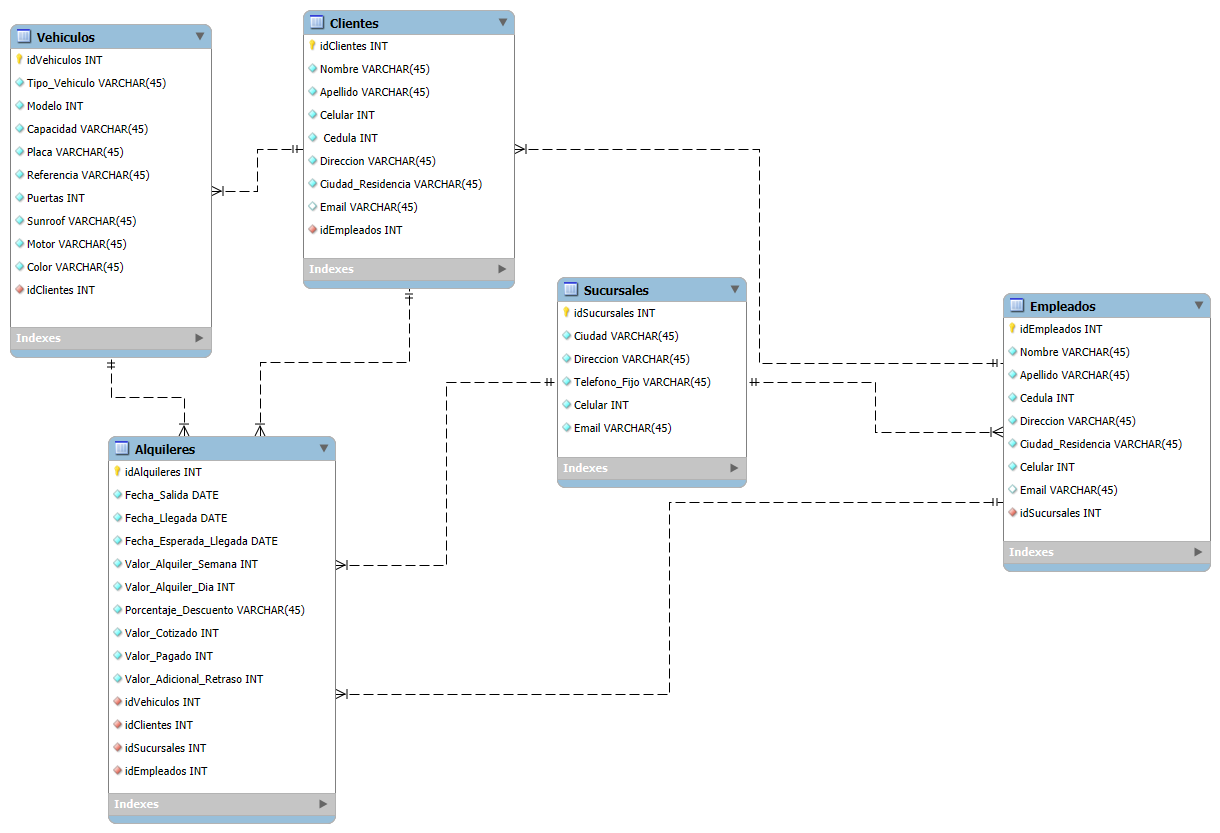
1. Tabla Sucursales:



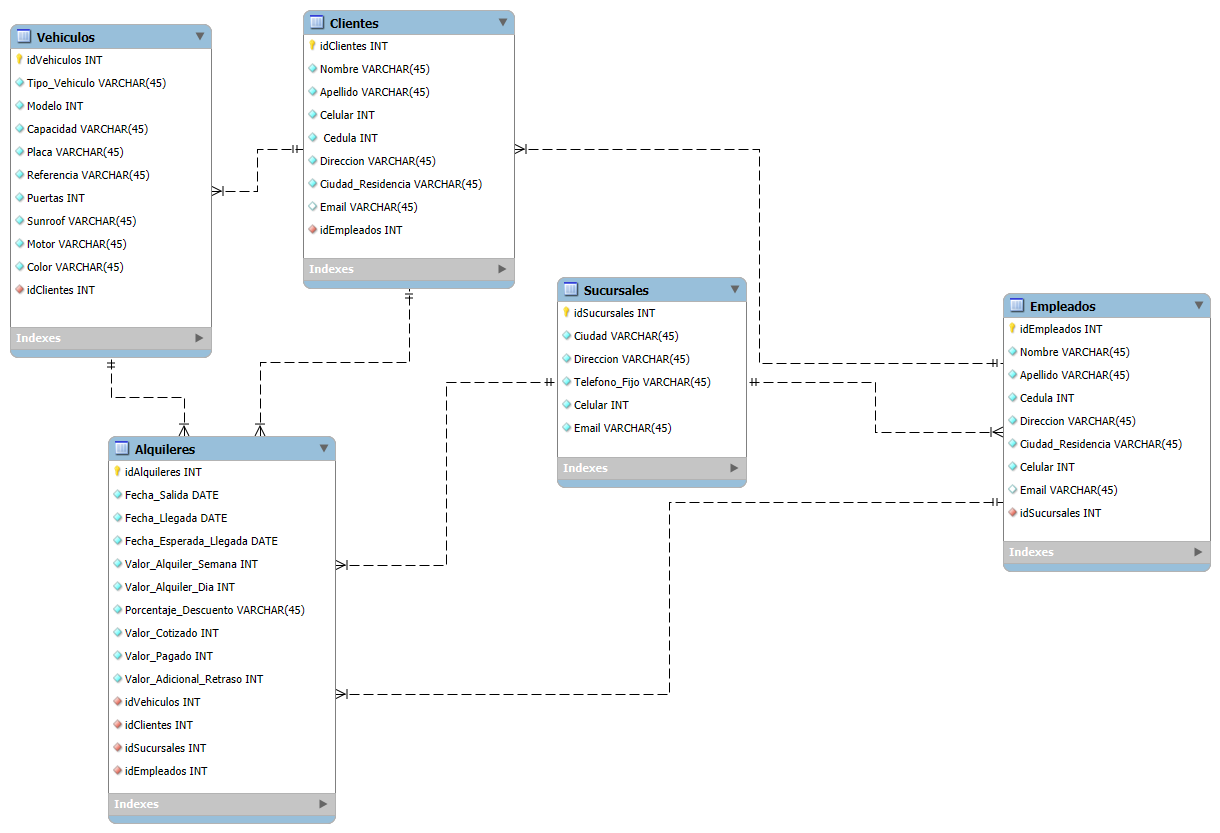
1. Tabla Empleados:



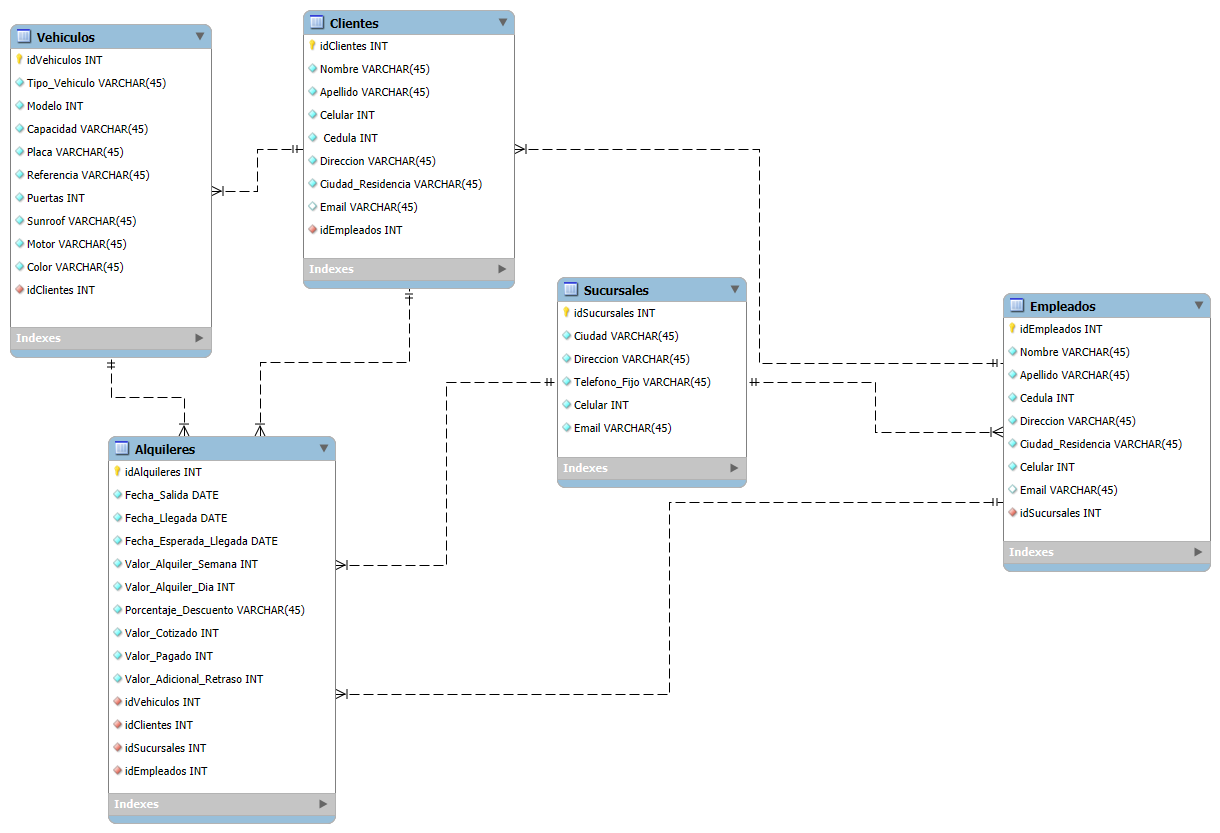
1. Tabla Clientes:



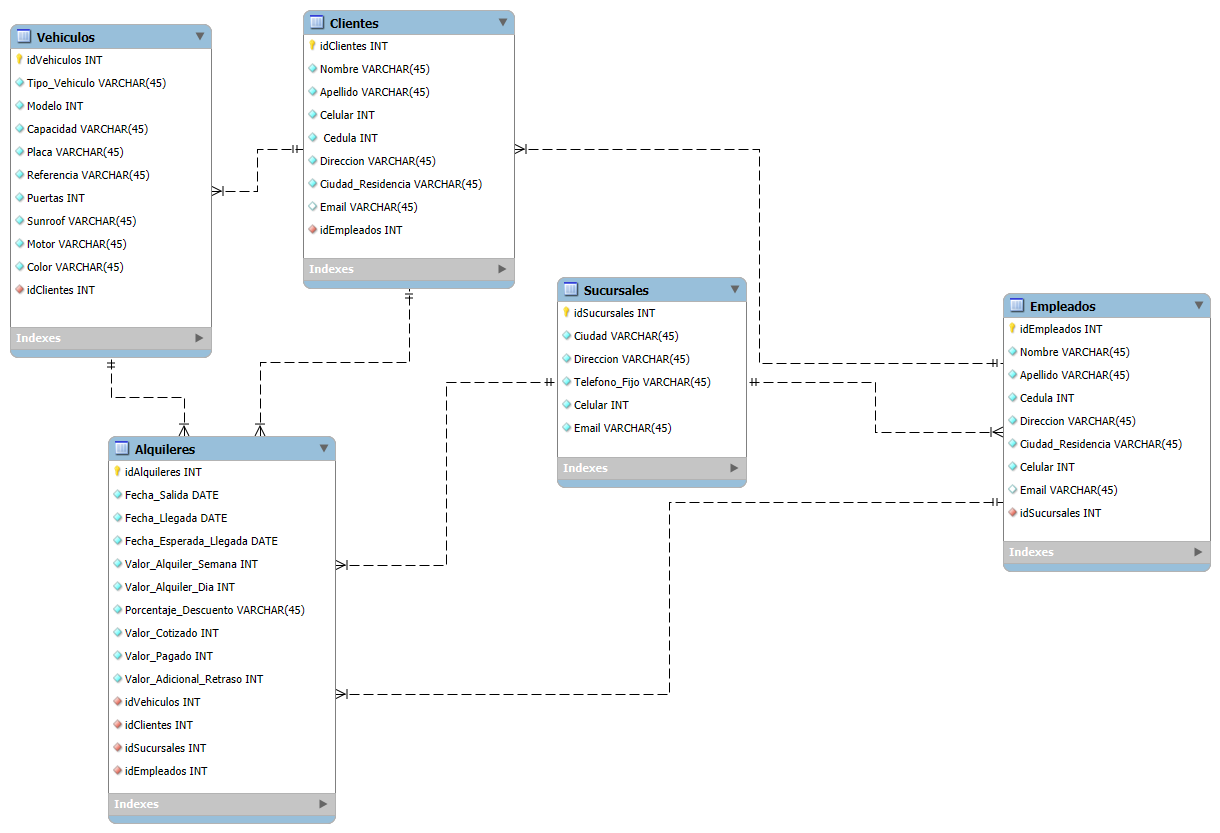
1. Tabla Vehículos:



1. Tabla Alquileres:



### Gráfica



El diagrama se subirá en formato PNG a GitHub con el resto del proyecto en la carpeta Dia8.