**MYSQL 2 - PLANEACIÓN COMPLETA BBDD**

# 

**JAIME ENRIQUE BARRERA SANDOVAL**

**T2**

**PEDRO FELIPE GÓMEZ BONILLA**

**CAMPUSLANDS**

**SANDBOX  
RUTA JAVA**

**TIBU**

**2024**

Tabla de Contenidos

[**Introducción 4**](#_heading=h.1fob9te)

[**Caso de Estudio 5**](#_heading=h.tyjcwt)

[**Planificación 5**](#_heading=h.3dy6vkm)

[Construcción del Modelo Conceptual 5](#_heading=h.1t3h5sf)

[Descripción 5](#_heading=h.2bn6wsx)

[Gráfica 5](#_heading=h.4d34og8)

[Descripción Técnica 6](#_heading=h.2s8eyo1)

[Construcción del Modelo Lógico 6](#_heading=h.17dp8vu)

[Descripción 6](#_heading=h.qsh70q)

[Gráfica 6](#_heading=h.3as4poj)

[Descripción Técnica 6](#_heading=h.3rdcrjn)

[Normalización del Modelo Lógico 6](#_heading=h.26in1rg)

[Primera Forma Normal (1FN) 7](#_heading=h.lnxbz9)

[Descripción 7](#_heading=h.1pxezwc)

[Gráfica 7](#_heading=h.49x2ik5)

[Descripción Técnica 7](#_heading=h.35nkun2)

[Segunda Forma Normal (2FN) 7](#_heading=h.1ksv4uv)

[Descripción 8](#_heading=h.2p2csry)

[Gráfica 8](#_heading=h.147n2zr)

[Descripción Técnica 8](#_heading=h.44sinio)

[Tercera Forma Normal (3FN) 8](#_heading=h.2jxsxqh)

[Descripción 8](#_heading=h.3o7alnk)

[Gráfica 8](#_heading=h.z337ya)

[Descripción Técnica 8](#_heading=h.3j2qqm3)

[Construcción del Modelo Físico 9](#_heading=h.1y810tw)

[Descripción 9](#_heading=h.4i7ojhp)

[Código 9](#_heading=h.2xcytpi)

[Descripción Técnica 10](#_heading=h.23ckvvd)

[Diagrama E-R 10](#_heading=h.3whwml4)

[Descripción 10](#_heading=h.ihv636)

[Gráfica 10](#_heading=h.32hioqz)

[Descripción Técnica 10](#_heading=h.1hmsyys)

[Tablas 11](#_heading=h.41mghml)

[Descripción 11](#_heading=h.2grqrue)

[Gráfica 11](#_heading=h.vx1227)

[Descripción Técnica 11](#_heading=h.3fwokq0)

[Relaciones entre Tablas 11](#_heading=h.1v1yuxt)

[Descripción 11](#_heading=h.4f1mdlm)

[Gráfica 12](#_heading=h.2u6wntf)

[Descripción Técnica 12](#_heading=h.19c6y18)

[Inserción de Datos 12](#_heading=h.3tbugp1)

[Descripción 12](#_heading=h.28h4qwu)

[Gráfica 12](#_heading=h.nmf14n)

[Descripción Técnica 12](#_heading=h.37m2jsg)

[**Referencias 13**](#_heading=h.3whwml4)

# 

# **Introducción**

En la actualidad, la gestión eficiente de información es esencial para cualquier empresa que alquila vehículos, en este caso AutoRenta ya que permite el manejo adecuado de los datos de las sucursales, empleados, clientes, vehículos y alquileres, una base de datos bien diseñada facilita el acceso a la información, mejora la toma de decisiones y optimiza proceso administrativos, este proyecto tiene como objetivo la construcción de un sistema de Gestión de Alquileres de Vehiculos AutoRenta utilizando un modelo de base de datos relacional, aplicando las mejores prácticas de diseño, normalización y desarrollo.

El sistema propuesto permitirá a la empresa almacenar información clave de la empresa sobre las sucursales, empleados, clientes, vehículos y alquileres. Además, se garantizará la integridad y consistencia de los datos mediante la implementación de técnicas de normalización asegurando que el modelo este optimizado para consultas rápidas y eficientes

# 

# **Caso de Estudio**

Consideramos que AutoRenta desea implementar un sistema digital para la gestión de la información. Actualmente, esta empresa maneja los registros de manera manual, lo que resulta en procesos ineficientes y propensos a errores. La empresa necesita una solución que le permita registrar, actualizar y consultar información sobre sus sucursales, empleados, clientes, vehículos y alquileres. el objetivo es diseñar una base de datos que modele esta situación, permitiendo la gestión de:

* **Sucursales**: Registro de datos de ciudad y dirección donde se ubica, teléfono fijo, celular y correo electrónico.
* **Empleados**: Información de sucursal donde labora, cédula, nombre, apellidos, dirección y ciudad de residencia, celular y correo electrónico.
* **Clientes**: Registro de datos personales de los clientes. Cédula, nombres, apellidos, dirección y ciudad de residencia, celular, correo electrónico.
* **Vehículos**: Datos de los vehículos que se imparten para alquileres. Tipo de vehículo, placa, referencia, modelo, puertas, capacidad, sunroof, motor, color.
* **Alquileres:** Información sobre el alquiler. 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.

el sistema debe facilitar el alquiler de vehículos y la asignación de los clientes a los vehiculos, tambien debe facilitar el alquiler en una sucursal y la entrega en otra sucursal, ofrecer descuentos sobre diferentes tipos de vehículos a los largo del año y cobros adicionales por los retrasos en las fechas de entrega con incremento del 8% en los días adicionales .

# **Planificación**

## **Construcción del Modelo Conceptual**

### El modelo conceptual es el primer paso en el diseño de una base de datos, donde se identifican las entidades principales, sus atributos y las relaciones entre ellas. En esta fase, no nos enfocamos en los detalles técnicos, sino en representar de manera abstracta la información que se gestionará en el sistema. El objetivo es tener una visión clara y comprensible de los elementos clave que formarán parte de la base de datos.

### **Descripción**

Se diseñó el modelo conceptual identificando cada una de las entidades, sus atributos y las relaciones entre ellas. Este modelo conceptual proporciona una visión clara y estructurada de cómo se organizan 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 Técnica**

1. **Las Entidades y Atributos**

* **Sucursales:**
  + id\_sucursal: id único de sucursal
  + ciudad: nombre de la ciudad
  + dirección: dirección dentro de la ciudad
  + teléfono\_fijo: Teléfono fijo de la sucursal
  + teléfono\_celular: Teléfono celular de la sucursal
  + correo\_electronico: Correo electrónico de la sucursal
* **Empleado:**
  + id\_empleado: id único de empleado
  + id\_sucursal (FK): Relación con la identidad sucursal
  + cédula: cédula de empleado
  + nombres: nombres del empleado
  + apellidos: apellidos del empleado
  + dirección: dirección del empleado dentro de la ciudad
  + ciudad\_residencia: nombre de la ciudad de residencia del empleado
  + celular: celular del empleado
  + correo\_electronico: correo electrónico del empleado
* **Cliente:**
  + id\_cliente: id único cliente
  + cédula: cédula cliente
  + nombres: nombres del cliente
  + apellidos: apellidos del empleado
  + dirección: dirección del cliente dentro de la ciudad
  + ciudad\_residencia: ciudad de residencia del cliente
  + celular: celular del cliente
  + correo\_electronico: correo del cliente
* **Vehículo:**
  + id\_vehiculo: id único del vehículo
  + tipo\_vehiculo: tipo de vehículo
  + placa: placa del vehículo
  + referencia: referencia del vehículo
  + modelo: modelo del vehículo
  + puertas: cantidad de puertas del vehículo
  + capacidad: capacidad del vehículo
  + sunroof (booleano): si el vehículo tiene techo corredizo o techo solar
  + motor: motor del vehículo
  + color: color del vehículo
* **Alquiler:**
  + id\_alquiler: id único del alquiler
  + id\_cliente (FK): Relación con la identidad cliente
  + id\_vehiculo (FK): Relación con la identidad vehículo
  + id\_empleado (FK): Relación con la identidad empleado
  + id\_sucursal\_salida (FK): Relación con la identidad sucursal
  + fecha\_salida: fecha de salida del vehículo cuando se alquila
  + id\_sucursal\_llegada (FK): Relación con la identidad sucursal
  + fecha\_llegada: fecha de llegada del vehículo cuando se alquila
  + fecha\_esperada\_llegada: fecha esperada de llegada del vehículo
  + valor\_alquiler\_por\_semana: valor de alquilar por semana
  + valor\_alquiler\_por\_dia: valor de alquiler por dia
  + porcentaje\_descuento: porcentaje de descuento
  + valor\_cotizado: valor cotizado aplicando descuento o sin descuento, lo mismo con adiciones
  + valor\_pagado: valor pagado por el alquiler

1. **Relaciones y Cardinalidades**

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

* **Vehículo - alquiler**
  + **Relación:** “es alquilado en”
  + **Descripción:** Un vehículo puede ser alquilado muchas veces pero cada alquiler involucra sólo un vehículo específico.
  + **Cardinalidad:** 1-N (uno a muchos)
* Empleado - alquiler
  + **Relación:** “gestiona”
  + **Descripción:**
  + **Cardinalidad:** 1-N (uno a muchos)

### **Gráfica**

## **Construcción del Modelo Lógico**

### El modelo lógico de una base de datos transforma el modelo conceptual en una estructura más detallada, donde se definen las claves primarias, claves foráneas, y las relaciones entre las tablas. En esta fase, se comienza a definir cómo se almacenarán los datos en la base de datos relacional, preservando la integridad y las relaciones establecidas en el modelo conceptual.

### **Descripción**

### El **Sistema** será modelado lógicamente con las siguientes tablas: **Vendedores**, **Vehículos**, **Ventas**, **Clientes y Mantenimiento** A través de estas tablas se mantendrán las relaciones y se definirán las claves necesarias para la integridad referencial.

### **Descripción Técnica**

**Entidad Vehículos**:

* Atributos:
  + **VIN**: Clave primaria, identifica de manera única cada vehículo.
  + **Modelo**: Modelo del Vehículo.
  + **Marca**: Marca del Vehículo.
  + **Estado**: Estado del Vehículo.
  + **Color**: Color del Vehículo.
  + **Año**: Año del Vehículo.
  + **Precio**: Precio del Vehículo.
  + **Tipo\_Combustible**: Combustible del Vehículo.
  + **Tipo\_Transmision**: Transmisión del Vehículo.

**Entidad Ventas**:

* Atributos:
  + **ID\_Venta**: Clave primaria, identifica de manera única cada venta.
  + **Metodo\_pago**: Método de pago.
  + **Total\_pago**: Pago total Vehículo.
  + **Fecha\_venta**: Fecha venta del Vehículo.
  + **Id\_cliente**: Clave foránea.
  + **Id\_vendedor**: Clave foránea.
  + **Id\_vehiculos**: Clave foránea.

**Entidad Clientes**:

* Atributos:
  + **ID\_Clientes**: Clave primaria, identifica de manera única a cada cliente.
  + **Nombre**: Nombre del cliente.
  + **Teléfono**: Teléfono del cliente.
  + **Dirección**: Dirección de cliente.
  + **Email**: Email de cliente.
  + **Id\_mantenimiento**: Clave foránea.
  + **Id\_vehiculos**: Clave foránea.

**Entidad Vendedores**:

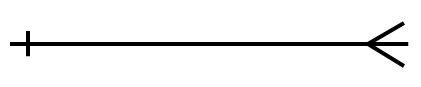
* Atributos:
  + **ID\_Vendedores**: Clave primaria, identifica de manera única a cada vendedor.
  + **Numero\_empleado**: Numero del empleado.
  + **Fecha\_contratacion**: Fecha de contratación del empleado.
  + **Id\_ventas**: Clave foránea.

**Entidad Mantenimiento** (Intermedia para relación N entre Estudiantes y Cursos):

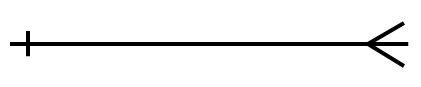
* Atributos:
  + **ID\_Mantenimiento**: Clave primaria de la tabla intermedia.
  + **Tipo\_Servicio:** Tipo de mantenimiento.
  + **Costo:** Costo del mantenimiento.
  + **Fecha\_Servicio:** Fecha que se realizó.
  + **ID\_Cliente**: Clave foránea.
  + **ID\_Vehiculo**: Clave foránea.

**Relaciones:**

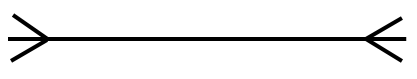
1. **Vehículos a Ventas**: Relación de **uno a muchos.**

****

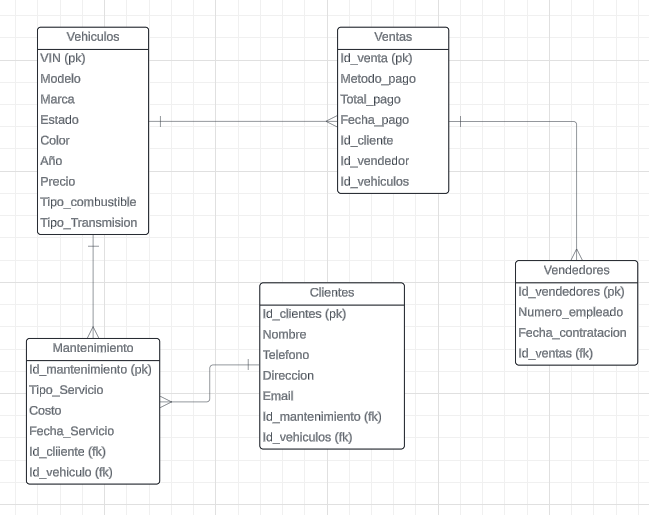
1. **Ventas a Vendedores**: Relación de **uno a muchos.**

****

1. **Vehículos** **a** **Clientes**: Relación de **muchos a muchos** gestionada por la tabla **Mantenimiento**.



### **Gráfica**



## **Normalización del Modelo Lógico**

La **normalización** es el proceso de estructurar las tablas de una base de datos para minimizar la redundancia y asegurar la integridad de los datos. Se realiza en varias fases llamadas **formas normales**. El proceso implica dividir las tablas en entidades más pequeñas y definir relaciones claras entre ellas para eliminar inconsistencias y dependencias innecesarias.

En el modelo lógico presentado, la normalización sigue los siguientes pasos:

### **Primera Forma Normal (1FN)**

#### La Primera Forma Normal (1FN) establece que una tabla debe estar estructurada de manera que cada columna solo contenga valores atómicos, es decir, valores que no pueden ser divididos, y cada campo de una fila debe contener un único valor. Este es el primer paso en el proceso de normalización, que garantiza que las tablas estén libres de grupos repetitivos y datos complejos.

#### **Descripción**

Para que una tabla esté en 1FN, debe cumplir con las siguientes reglas:

1. **No debe haber grupos repetidos**: Cada columna debe contener un único valor y no listas o subconjuntos de valores.
2. **Cada campo debe contener un solo valor**: No puede haber múltiples valores en una misma celda.
3. **Cada registro debe ser único**: Debe haber una clave primaria que identifique de manera única cada fila de la tabla.

En este modelo, la tabla **Vendedores** debe estar en 1FN eliminando cualquier dato que no sea atómico, como listas de nombres o fechas múltiples en una sola columna.

#### **Gráfica**

| Id\_Vendedores (pk) | Numero\_empleado | Fecha\_contratacion | Id\_ventas |
| --- | --- | --- | --- |
| 1 | 1458 | 2000-01-10 | 1 |
| 2 | 5478 | 2008-05-12 | 2 |

En la tabla anterior:

* Cada celda contiene un valor único y atómico.
* La columna **Id\_Vendedores** es la clave primaria que asegura la unicidad de cada registro.

#### **Descripción Técnica**

**Aplicación de 1FN en el Modelo**

* Si en la tabla **Vendedores** se almacenan múltiples valores en una sola celda (por ejemplo, varios números o fechas), esta tabla no estaría en 1FN.
* Se reorganiza la tabla para que cada celda contenga un único valor, y se utiliza una clave primaria para asegurar que cada fila es única.

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

#### La Segunda Forma Normal (2FN) es el segundo paso en la normalización de bases de datos. Una tabla está en 2FN si ya cumple con la Primera Forma Normal (1FN) y, además, todos los atributos no clave dependen completamente de la clave primaria. Esto significa que no puede haber dependencias parciales, lo que ocurre cuando un atributo no clave depende solo de una parte de una clave primaria compuesta.

#### **Descripción**

Para que una tabla esté en 2FN, debe cumplir con dos reglas principales:

1. **Cumplir con la Primera Forma Normal (1FN)**: La tabla no debe tener valores repetidos o no atómicos.
2. **Eliminación de dependencias parciales**: Todos los atributos no clave deben depender completamente de toda la clave primaria. Si la clave primaria es compuesta, no puede haber atributos que dependan solo de una parte de la clave.

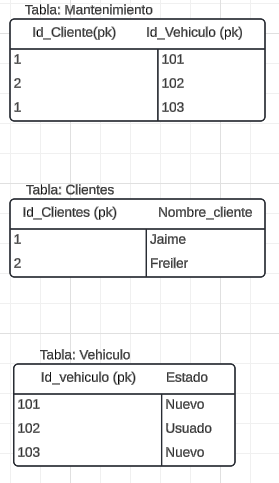
Esto es particularmente relevante en tablas donde se usa una clave primaria compuesta (formada por más de una columna), como en la tabla **Mantenimiento**, que relaciona Vehiculos con Clientes.

#### **Gráfica**

Antes de aplicar 2FN (Dependencias Parciales)

| Id\_cliente (pk) | Id\_vehiculo (pk) | Marca | Nombre\_Cliente |
| --- | --- | --- | --- |
| 1 | 101 | Ford | Jaime |
| 2 | 102 | Suzuki | Freiler |
| 1 | 103 | Mazda | Jaime |

Después de aplicar 2FN (Eliminación de Dependencias Parciales)

  
  
Aquí se han creado tablas separadas para **Clientes** y **Vehiculos**, eliminando la dependencia parcial. Ahora, el **Nombre\_Cliente** y el **Marca** se almacenan en sus respectivas tablas, y la tabla **Mantenimiento** solo contiene las claves primarias de ambas.

#### **Descripción Técnica**

1. **Dependencias parciales**: Ocurren cuando un atributo no clave en una tabla con clave compuesta depende solo de una parte de esa clave.
2. **Eliminación de dependencias parciales**: Para alcanzar la 2FN, los atributos que dependen parcialmente de una clave compuesta deben moverse a tablas separadas.
3. **Claves primarias compuestas**: En tablas como **Mantenimiento**, la clave primaria está formada por dos columnas (ID\_Vehiculo y ID\_Cliente). Todos los atributos no clave deben depender de ambas columnas de manera conjunta.
4. **Tabla Intermedia**: La tabla **Mantenimiento** actúa como una tabla intermedia que conecta a los Vehiculos con los clientes, y ya no contiene atributos innecesarios.

**Aplicación de 2FN en el Modelo**

* La clave primaria compuesta en la tabla **Mantenimiento** (formada por ID\_Vehiculo y ID\_Cliente) asegura que no haya dependencias parciales.
* Los datos que dependen de una sola columna de la clave primaria (como el Marca o Nombre\_cliente) se almacenan en tablas separadas, garantizando que la tabla intermedia sea eficiente y no repita información innecesaria.

### **Tercera Forma Normal (3FN)**

#### La Tercera Forma Normal (3FN) es una etapa avanzada en la normalización de bases de datos que tiene como objetivo eliminar las dependencias transitivas. Una tabla está en 3FN cuando cumple con la Segunda Forma Normal (2FN) y, además, ningún atributo no clave depende de otro atributo no clave, es decir, los atributos no clave solo deben depender directamente de la clave primaria.

#### **Descripción**

Para que una tabla esté en 3FN, debe cumplir con las siguientes reglas:

1. **Cumplir con la Segunda Forma Normal (2FN)**: La tabla no debe tener dependencias parciales.
2. **Eliminar dependencias transitivas**: No debe haber dependencias entre los atributos no clave. Todos los atributos no clave deben depender únicamente de la clave primaria.

Una **dependencia transitiva** ocurre cuando un atributo no clave depende de otro atributo no clave, que a su vez depende de la clave primaria. Este tipo de dependencia introduce redundancia en los datos y debe eliminarse para que el modelo esté en 3FN.

#### **Gráfica**

Antes de aplicar 3FN (Dependencia Transitiva)

| Id\_cliente (pk) | Estado | Marca | Nombre\_Cliente |
| --- | --- | --- | --- |
| 1 | Nuevo | Ford | Jaime |
| 2 | Usado | Suzuki | Freiler |
| 3 | Nuevo | Mazda | Jaime |

En esta tabla, **Nombre\_Cliente** depende de la **Marca**, no directamente de la **ID\_Cliente**, lo que es una dependencia transitiva. Esto crea redundancia porque si cambiara el nombre de un cliente, se tendría que actualizar en varias filas.  
  

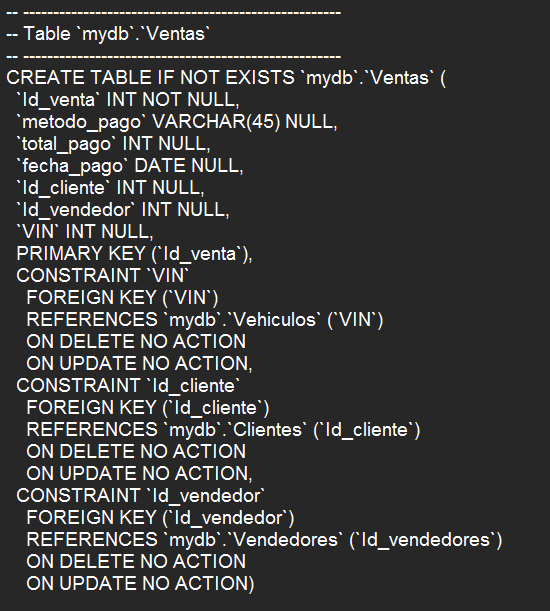

Se ha creado una nueva tabla llamada **Marca** para almacenar la **Ubicación**, eliminando la dependencia transitiva de la tabla **Nombre\_cliente**.

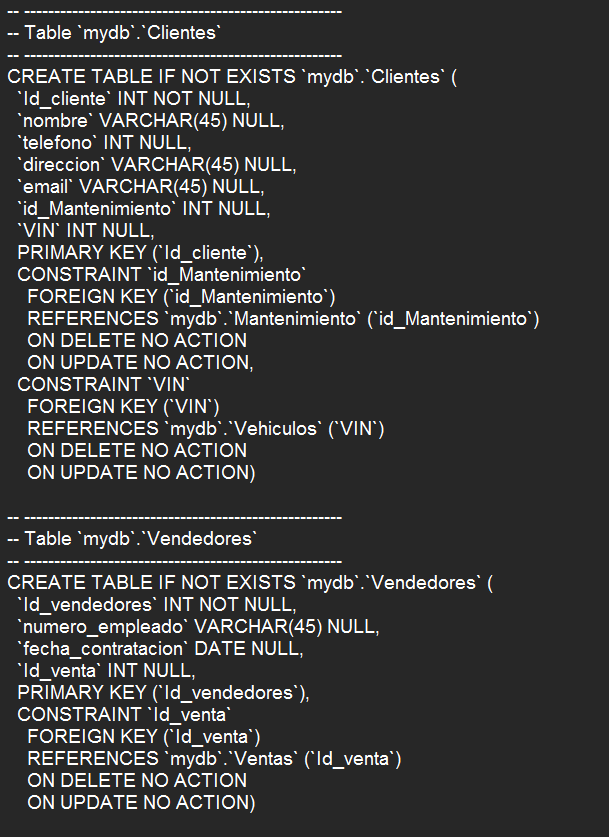
#### **Descripción Técnica**

1. **Dependencias transitivas**: Ocurren cuando un atributo no clave depende de otro atributo no clave.
2. **Eliminación de dependencias transitivas**: Para alcanzar la 3FN, los atributos que dependen de otro atributo no clave deben moverse a una tabla separada.
3. **Clave foránea**: Después de la normalización, la tabla **Clientes** puede hacer referencia al marca mediante una clave foránea, lo que garantiza la integridad referencial.
4. **Optimización de la base de datos**: La 3FN reduce redundancias y actualizaciones innecesarias, lo que mejora el rendimiento y facilita el mantenimiento de la base de datos. Al eliminar dependencias transitivas, los datos se estructuran de manera más eficiente.

## **Construcción del Modelo Físico**

### **Descripción**

Un modelo de datos físico es un modelo específico de bases de datos que representa objetos de datos relacionales, por ejemplo, tablas, columnas, claves principales y claves externas y sus relaciones.  
  
**Código**



### **Construcción del diagrama UML**

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

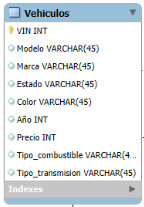
mejor los diseños, la arquitectura del código y la implementación propuesta. Este enfoque nos permitirá tener una visión clara y detallada de cómo se manejan cada una de las consultas,funcionalidades y los usuarios en la base de datos. De esta manera, podremos asegurar una implementación eficiente y coherente con los requisitos del sistema.

**Descripción**

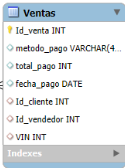
El diagrama UML se ha diseñado con el objetivo de representar detalladamente la estructura de cada tabla y sus relaciones. Este diagrama ilustra claramente el tipo de dato correspondiente a cada atributo, así como la identificación de claves primarias (primary keys) y claves foráneas (foreign keys). 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.

### **Tablas:**

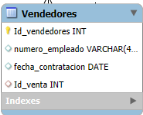
* **Vehículos**: Registro de datos como marca, modelo y VIN.



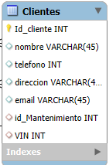
* **Ventas**: Información sobre los vehículos ofrecidos por la empresa.



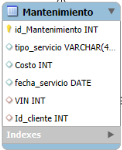
* **Vendedores**: Datos de los vendedores que imparten las ventas.



* **Clientes**: Registro de datos personales de los clientes

.

* **Mantenimiento** (Intermedia para relación N entre Estudiantes y Cursos)



**Gráfica**

