**LIBRERÍA DEL MUNDO DE SOFIA**

**MARIA GUADALUPE LIZARAZO LEAL**

**T2**

**PEDRO FELIPE GÓMEZ BONILLA**

**CAMPUSLANDS**

**SANDBOX RUTA JAVA**

**FLORIDABLANCA**

**2024**

**Tabla de contenido**

[INTRODUCCIÓN 3](#_Toc181176660)

[Caso de Estudio 4](#_Toc181176661)

[Planificación 5](#_Toc181176662)

[Construcción del Modelo Conceptual 5](#_Toc181176663)

[Descripción 5](#_Toc181176664)

[Gráfica 8](#_Toc181176665)

[Construcción del Modelo Lógico 8](#_Toc181176666)

[Descripción 8](#_Toc181176667)

[Gráfica 11](#_Toc181176668)

[Normalización del Modelo Lógico 11](#_Toc181176669)

[Primera Forma Normal (1FN) 11](#_Toc181176670)

[Segunda Forma Normal (2FN) 14](#_Toc181176671)

# INTRODUCCIÓN

Este proyecto filtro está basado en una librería la cual tiene ciertos requerimientos previamente solicitados, los cuales servirán para la calificación final de la materia introducción al Backend. Mostrando así los conceptos aprendidos de BBDD.

A continuación, analizaremos la tienda de libros del mundo de Sofía, para la cual se diseñará una base de datos que nos permita gestionar el inventario, las ventas y los clientes. Esta base de datos deberá permitir el registro y la gestión de los libros, autores, clientes, pedidos y transacciones de compra.

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 tienda de libros Sofía, nos ha planteado un informe del cual nos basaremos, analizando y poniendo énfasis en todos los requerimientos y solicitudes.

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

* **Libros**: Un libro tiene un título, uno o más autores, editorial, categoría, fecha de publicación, ISBN único, precio y cantidad en stock. Permitir buscar libros por título, autor, categoría o ISBN.
* **Autores**: Un autor tiene un nombre, fecha de nacimiento y nacionalidad. Un libro puede tener varios autores y un autor puede haber escrito varios libros.
* **Clientes**: Los clientes se registran con nombre, correo electrónico, teléfono y dirección. Un cliente puede realizar varios pedidos.
* **Pedidos**: Un pedido puede incluir múltiples libros, con cantidades específicas. Asociar cada pedido a un cliente y una fecha de compra. Gestionar el estado del pedido (pendiente, procesado, completado).
* **Transacciones**: Cada pedido genera una transacción con el método de pago (tarjeta de crédito, PayPal, etc.), el monto total y la fecha de la transacción.
* **Relaciones y Restricciones**: Implementar relaciones entre libros y autores, entre clientes y pedidos, y entre pedidos y transacciones. Evitar registros duplicados de libros. La cantidad en stock de un libro debe actualizarse al realizar una compra.

# 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. **Clientes**:

* Id\_Clientes: identificación única por cliente.
* Nombre\_Cliente: nombres del cliente.
* Apellido\_Cliente: apellidos del cliente.
* Telefono\_Cliente: número de contacto del cliente.
* Dirección\_Cliente: dirección de residencia del cliente.
* Email\_Cliente: correo electrónico de contacto con el cliente.

1. **Pedidos**:

* Id\_Pedidos: identificación única por pedido.
* Fecha\_Pedido: fecha en la que se realizó el pedido.
* Hora\_Pedido: hora en la que se realizó el pedido.
* Cantidad\_Pedido: cantidad de libros que solicitaron en el pedido.
* Estado\_Pedido: estado en el que se encuentra el pedido.

1. **Ventas**:

* Id\_Ventas: identificación única por venta.
* Fecha\_Venta: fecha en la que se realizó la venta.
* Hora\_Venta: hora en la que se realizó la venta.
* Precio\_Total: monto final a cobrar por la venta.
* Método\_ Pago: opciones en la que se puede cancelar la venta.
* Factura\_Venta: soporte final de la venta.

1. **Transacciones**:

* Id\_Transacción: identificación única por cada transacción realizada.
* Fecha\_transferencia: fecha en la que se realizó la transferencia.
* Forma\_pago: opción de bancos para la transacción.
* Monto\_transferido: precio final a transferir.

1. **Libros**:

* ISBN: código único por libro.
* Autor: nombre del autor del libro.
* Título: nombre o título del libro.
* Categoría: tipo de libro.
* Fecha\_publicación: fecha en la que se publicó el libro.
* Precio: precio del libro.
* Editorial: lugar que publicó el libro.
* Cantidad: cantidad de libros que haya.

1. **Autores**:

* Id\_autor: identificación única de cada autor.
* Nombre\_autor: nombres del autor.
* Apellido\_autor: apellidos del autor.
* Fecha\_nacimiento: fecha en la que nació el autor.
* Nacionalidad\_autor: país natal del autor.

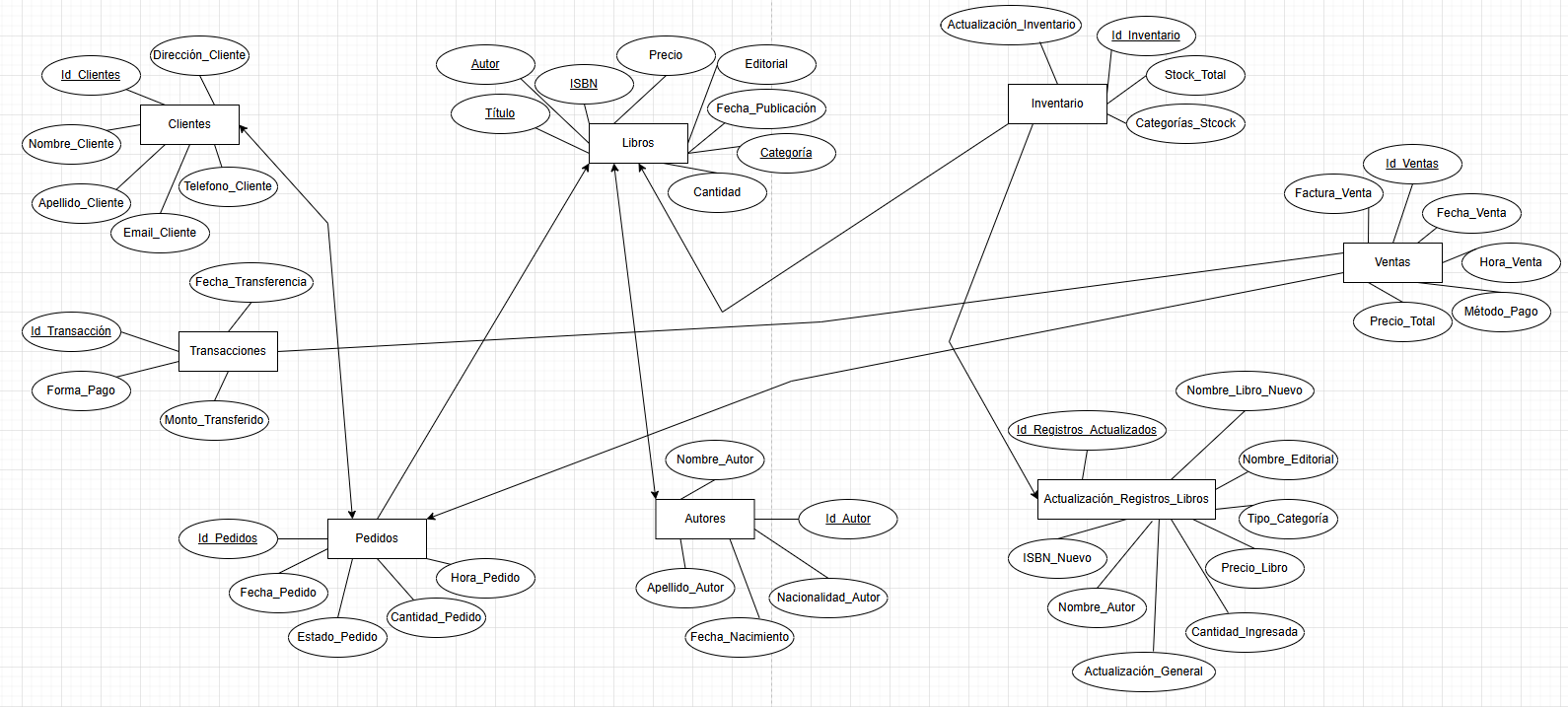
1. **Inventario**:

* Id\_inventario: identificación única del inventario.
* Actualización\_inventario: actualización real del inventario.
* Stock\_total: stock total de la tienda.
* Categorías\_stock: cantidad de categorías en stock.

1. **Actualización\_Registros\_Libros**:

* Id\_Registros\_Actualizados: identificación única de cada registro.
* ISBN\_Nuevo: código nuevo y único de cada libro.
* Nombre\_Libro\_Nuevo: título del nuevo libro.
* Nombre\_Autor: nombre del autor del libro.
* Nombre\_Editorial: lugar que publicó el libro.
* Tipo\_Categoría: opción de categorías de los libros.
* Cantidad\_Ingresada: cantidad de libros a ingresar.
* Precio\_Libro: precio de compra del libro.
* Actualización\_General

### 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. **Clientes**:

* Id\_Clientes: INT PRIMARY KEY.
* Nombre\_Cliente: VARCHAR(15) NOT NULL.
* Apellido\_Cliente: VARCHAR(15) NOT NULL.
* Telefono\_Cliente: INT NOT NULL.
* Dirección\_Cliente: VARCHAR(20) NOT NULL.
* Email\_Cliente: VARCHAR(20) NOT NULL.

1. **Clientes\_has\_Pedidos**:

* Clientes\_idClientes: INT FOREIGN KEY NOT NULL.
* Pedidos\_idPedidos: INT FOREIGN KEY NOT NULL.

1. **Pedidos**:

* Id\_Pedidos: INT PRIMARY KEY.
* Fecha\_Pedido: INT NOT NULL.
* Hora\_Pedido: INT NOT NULL.
* Cantidad\_Pedido: INT NOT NULL.
* Estado\_Pedido: VARCHAR(20) NOT NULL.

1. **Ventas**:

* Id\_Ventas: INT PRIMARY KEY.
* Fecha\_Venta: INT NOT NULL.
* Hora\_Venta: INT NOT NULL.
* Precio\_Total: INT NOT NULL.
* Método\_ Pago: VARCHAR(20) NOT NULL.
* Factura\_Venta: VARCHAR(45) NOT NULL.

1. **Transacciones**:

* Id\_Transacción: INT PRIMARY KEY.
* Fecha\_transferencia: INT NOT NULL.
* Forma\_pago: VARCHAR(15) NOT NULL.
* Monto\_transferido: INT NOT NULL.

1. **Libros**:

* ISBN: INT PRIMARY KEY
* Autor: VARCHAR(20) NOT NULL.
* Título: VARCHAR(20) NOT NULL.
* Categoría: VARCHAR(20) NOT NULL.
* Fecha\_publicación: INT NOT NULL.
* Precio: INT NOT NULL.
* Editorial: VARCHAR(20) NOT NULL.
* Cantidad: INT NOT NULL.

1. **Libros\_has\_Autores**:

* Libros\_ISBN: INT FOREIGN KEY NOT NULL.
* Libros\_Título: INT FOREIGN KEY NOT NULL.
* Libros\_Autor: INT FOREIGN KEY NOT NULL.
* Libros\_Categoria: INT FOREIGN KEY NOT NULL.
* Autores\_idAutores: INT FOREIGN KEY NOT NULL.

1. **Autores**:

* Id\_autor: INT PRIMARY KEY.
* Nombre\_autor: VARCHAR(15) NOT NULL.
* Apellido\_autor: VARCHAR(15) NOT NULL.
* Fecha\_nacimiento: INT NOT NULL.
* Nacionalidad\_autor: VARCHAR(20) NOT NULL.

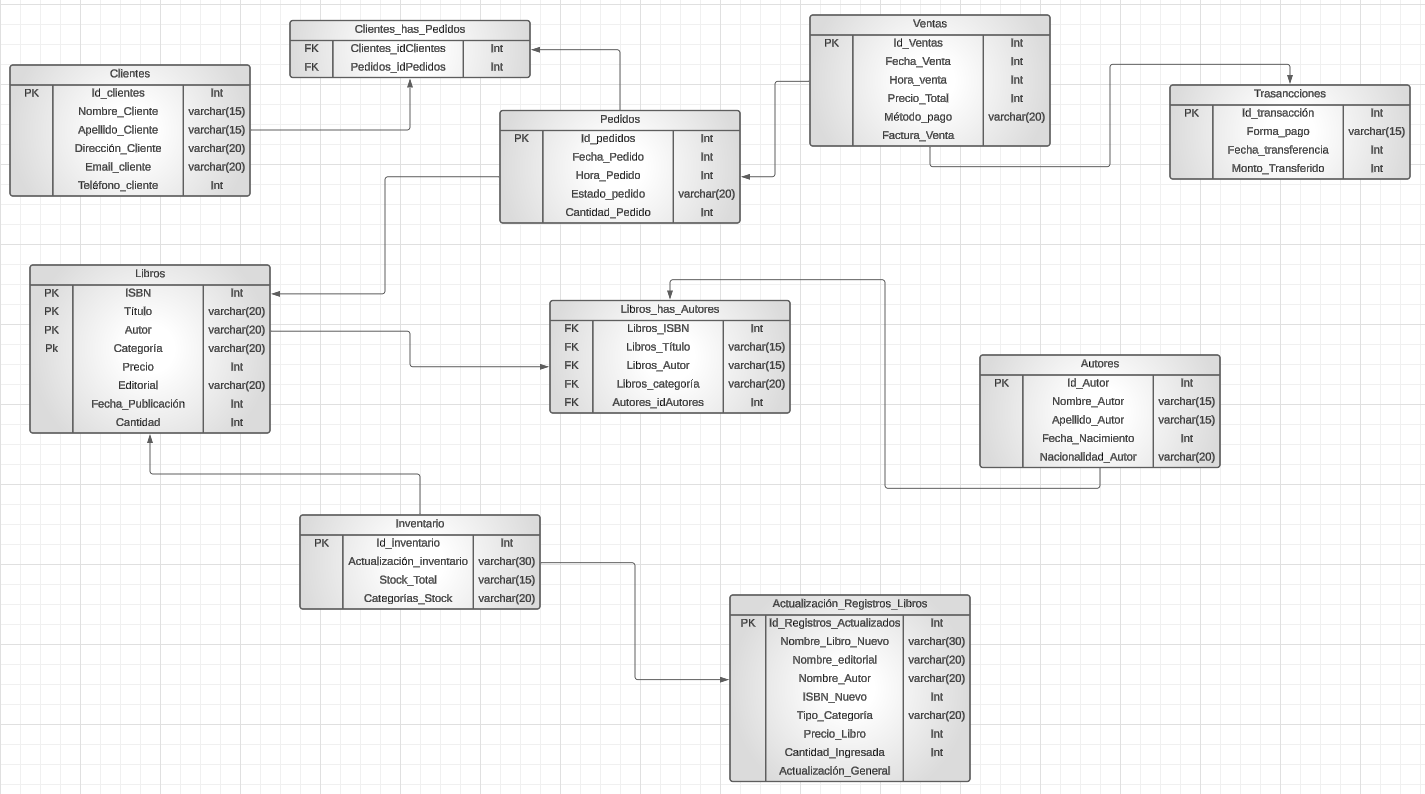
1. **Inventario**:

* Id\_inventario: INT PRIMARY KEY.
* Actualización\_inventario: VARCHAR(30) NOT NULL.
* Stock\_total: VARCHAR(15) NOT NULL.
* Categorías\_stock: VARCHAR(20) NOT NULL.

1. **Actualización\_Registros\_Libros**:

* Id\_Registros\_Actualizados: INT PRIMARY KEY.
* ISBN\_Nuevo: INT NOT NULL.
* Nombre\_Libro\_Nuevo: VARCHAR(30) NOT NULL.
* Nombre\_Autor: VARCHAR(20) NOT NULL.
* Nombre\_Editorial: VARCHAR(20) NOT NULL.
* Tipo\_Categoría: VARCHAR(20) NOT NULL.
* Cantidad\_Ingresada: INT NOT NULL.
* Precio\_Libro: INT NOT NULL.
* Actualización\_General: VARCHAR(45) 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. **Clientes\_has\_Pedidos**:

* 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. **Pedidos**:

* 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. **Ventas**:

* 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. **Transacciones**:

* 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. **Libros**:

* 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. **Libros\_has\_Autores**:

* 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. **Autores**:

* 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.

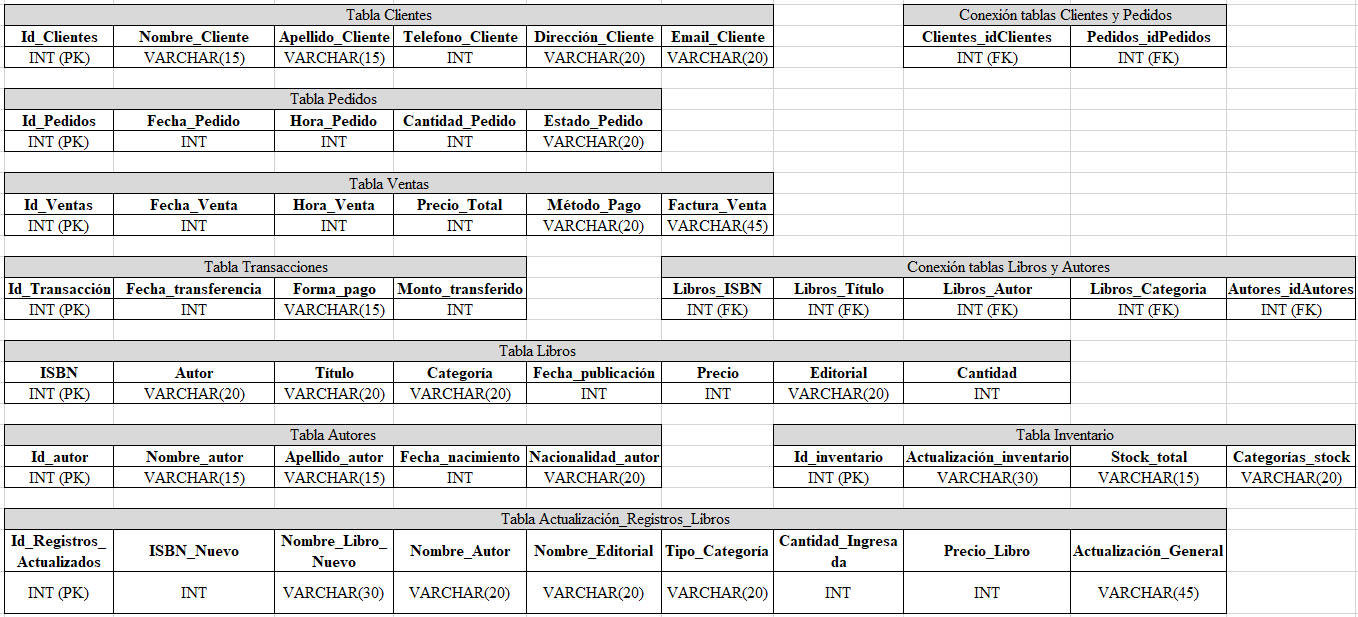
1. **Inventario**:

* 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. **Actualización\_Registros\_Libros**:

* 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. **Clientes\_has\_Pedidos**:

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

1. **Pedidos**:

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

1. **Ventas**:

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

1. **Transacciones**:

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

1. **Libros**:

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

1. **Libros\_has\_Autores**:

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

1. **Autores**:

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

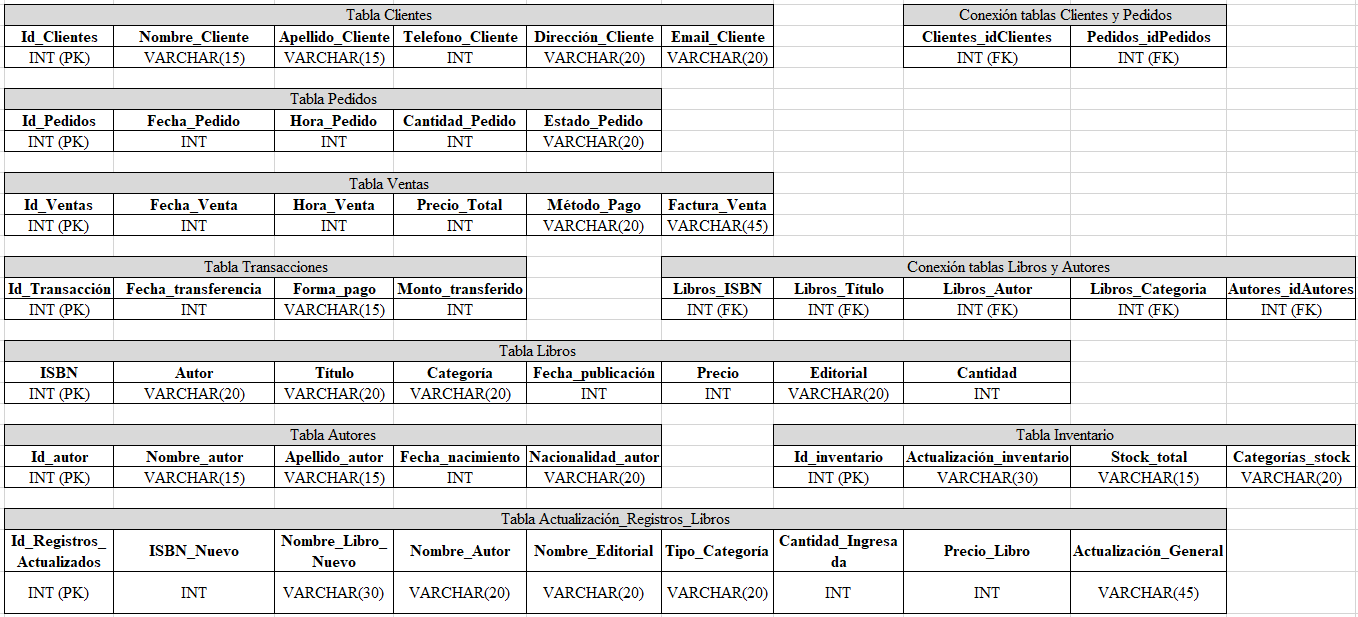
1. **Inventario**:

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

1. **Actualización\_Registros\_Libros**:

* 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 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 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. **Clientes\_has\_Pedidos**:

* 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. **Pedidos**:

* 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. **Ventas**:

* 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. **Transacciones**:

* 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. **Libros**:

* 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. **Libros\_has\_Autores**:

* 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. **Autores**:

* 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.

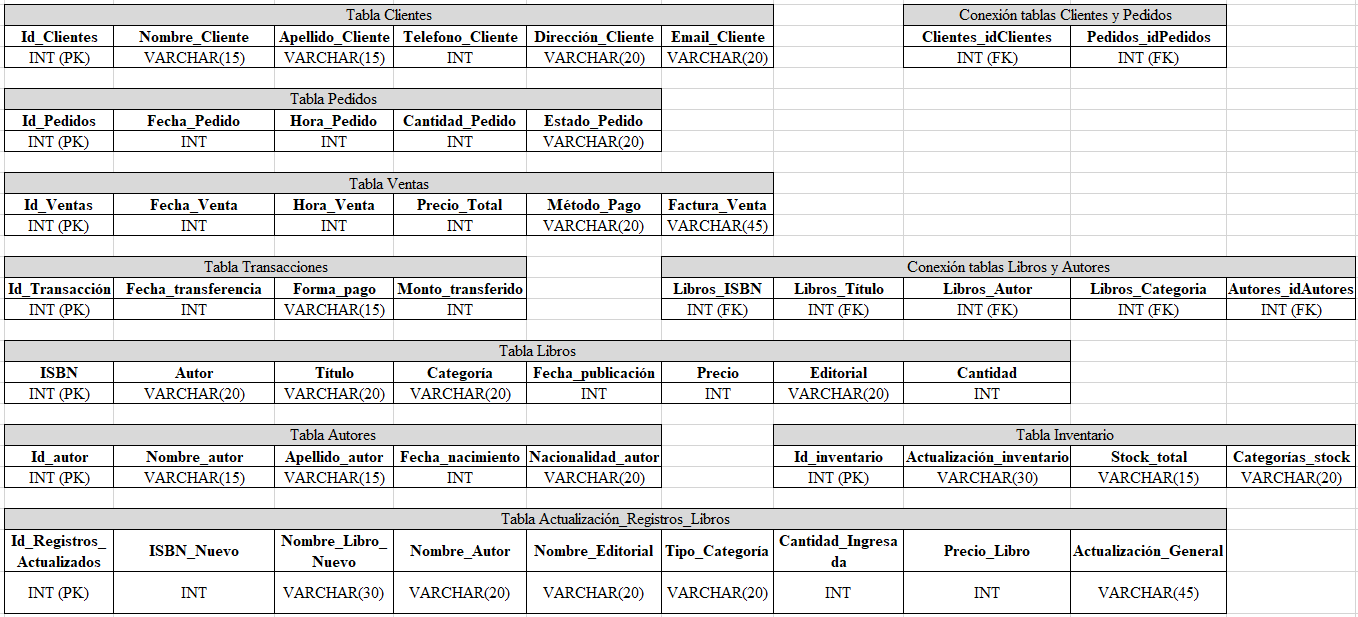
1. **Inventario**:

* 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. **Actualización\_Registros\_Libros**:

* 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: