TAREA: ESTABLECER UNA CONEXIÓN CON UNA BASE DE DATOS PARA DISPOSITIVOS MÓVILES

DIEGO JURADO GUEVARA

YAIDER CAICEDO

### INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO DEPARTAMENTO DE ELECTRÓNICA TECNOLOGÍA EN DESARROLLO DE SOFTWARE MEDELLÍN

### 2023

**Caso de estudio**

**Conexión Y Diseño De Bases De Datos Práctica De Laboratorio Guía De Apoyo**

La siguiente práctica se denomina, conexión y diseño de bases de datos. Esta consiste en la creación del modelo entidad relación para la ferretería rodamientos y fierros. La información que se genera en la ferretería está relacionada con los clientes de la ferretería, de cada cliente se desea conocer el código, el nombre, la dirección y el teléfono. Cada cliente puede realizar varios pedidos y un pedido solo lo puede realizar un solo cliente, de cada pedido se desea conocer el código del pedido la descripción y la fecha del pedido. De los productos que la ferretería ofrece a sus clientes, se debe conocer el código del producto, el fabricante y el valor del producto. De cada factura que se genera por la compra de productos se debe conocer el código de la factura, la fecha y el valor de la factura.

La práctica consiste en crear el modelo entidad relación que responda a los requerimientos planteados anteriormente. Luego de crear el modelo entidad relación, se debe normalizar el modelo, hasta su tercera forma normal.

Excelente, basándome en esa información, puedo esbozar un modelo entidad-relación inicial:

Entidades:

- Cliente (Código, Nombre, Dirección, Teléfono)

- Pedido (Código del Pedido, Descripción, Fecha del Pedido)

- Producto (Código del Producto, Fabricante, Valor)

- Factura (Código de Factura, Fecha, Valor)

Relaciones:

- Cliente realiza Pedido (1:N) - Un cliente puede realizar varios pedidos.

- Pedido está relacionado con Producto (N:M) - Un pedido puede contener varios productos y un producto puede estar en varios pedidos.

- Factura se relaciona con Pedido (1:1) - Cada factura se asocia con un solo pedido.

Normalizar el modelo significa eliminar la redundancia y las dependencias para asegurar que esté en la tercera forma normal (3FN). Sigamos este proceso:

**1. Primera Forma Normal (1FN):**

- Asegurar que cada celda en la tabla contenga un solo valor, evitando valores repetidos.

- Crear tablas separadas para elementos multivaluados.

**2. Segunda Forma Normal (2FN):**

- Todos los atributos no clave deben depender completamente de la clave primaria.

- Si hay dependencias parciales, dividir las tablas.

**3. Tercera Forma Normal (3FN):**

- No deben existir dependencias transitivas.

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

**Cliente (Código, Nombre, Dirección, Teléfono)** está en 1FN ya que cada atributo tiene un solo valor.

**Pedido (Código del Pedido, Descripción, Fecha del Pedido**) está en 1FN por la misma razón.

**Producto (Código del Producto, Fabricante, Valor**) también está en 1FN.

**Factura (Código de Factura, Fecha, Valor**) también cumple con 1FN.

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

En este punto, asegurémonos de que cada atributo no clave esté completamente relacionado con la clave primaria.

En la relación Cliente realiza Pedido (1:N):

- La tabla Pedido ya está en 2FN, ya que la información está directamente relacionada con el pedido.

- En Cliente, la información relevante es solo el Código del Cliente, por lo que no hay dependencias parciales.

En la relación Pedido está relacionado con Producto (N:M):

- Parece que hay una relación N:M, lo que sugiere la necesidad de una tabla intermedia que relacione Pedidos y Productos.

En la relación Factura se relaciona con Pedido (1:1):

- Parece estar bien en cuanto a 2FN, ya que la Factura se relaciona directamente con el Pedido.

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

En la tabla Cliente, no parece haber dependencias transitivas, ya que todas las columnas están directamente relacionadas con la clave primaria.

En la tabla Producto, tanto el Fabricante como el Valor parecen depender directamente del Código del Producto, por lo que no hay dependencias transitivas.

En la relación entre Pedido y Producto (N:M), si hay atributos específicos del pedido relacionados con el producto, es posible que se deba descomponer esa relación en una tabla independiente para eliminar cualquier posible dependencia transitiva.

Con esto en mente, se podría estructurar el modelo normalizado de la siguiente manera:

- Cliente (Cedula, Nombre, Dirección, Teléfono)

- Pedido (Código ,  **Nit\_usuario**, Descripción, Fecha, **Anulado** )

- Producto (Código , Descripcion, Valor)

- pedidoLinea (Código , Código del Pedido, Cantidad , Codigo\_producto)

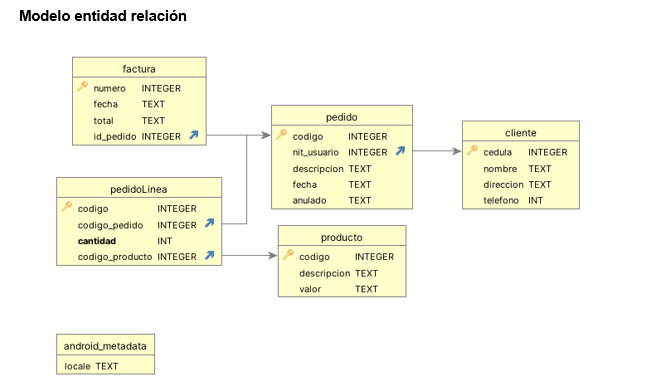
- Factura (Numero, Fecha, Total , Id\_Pedido)

.

# Script SQL para crear las tablas normalizadas en SQLite

|  |
| --- |
| Creación de la tabla de clientesCREATE TABLE Cliente (Cedula INT PRIMARY KEY,Nombre TEXT,Direccion TEXT,Telefono INT);Creación de la tabla de pedidoCREATE TABLE Pedido (Codigo INT PRIMARY KEY,Nit\_usuario INT,Descripcion TEXT,Fecha TEXT,Anulado TEXT,FOREIGN KEY (Nit\_usuario) REFERENCES Cliente(Cedula));Creación de la tabla de productosCREATE TABLE Producto (Codigo INT PRIMARY KEY,Descripcion TEXT,Valor TEXT);Tabla pedidoLineaCREATE TABLE pedidoLinea (Codigo INT PRIMARY KEY ,Codigo\_pedido INT,Cantidad INT,Codigo\_producto INT,FOREIGN KEY (Codigo\_Pedido) REFERENCES Pedido(Codigo),FOREIGN KEY (Codigo\_Producto) REFERENCES Producto(Codigo));Creación de la tabla de facturasCREATE TABLE Factura (Numero INT PRIMARY KEY,Fecha TEXT,Total TEXT ,Id\_Pedido INT,FOREIGN KEY (Id\_Pedido) REFERENCES Pedido(Id\_Pedido)); |

# Modelo entidad relación



**Link Repositorio**

<https://github.com/yadier2/ConexionDBSqlite>