

## 1. Introducción

El objetivo del diseño es construir una base de datos relacional que gestione clientes, productos y órdenes de compra de manera eficiente, evitando redundancias y asegurando la integridad de los datos. Se aplicaron las reglas de normalización hasta la Tercera Forma Normal (3FN), garantizando que cada tabla represente una sola entidad y que los atributos dependan únicamente de la clave primaria.

## 2. Proceso de Normalización

### 1FN (Primera Forma Normal)

- Todos los atributos son atómicos (no se almacenan listas ni valores compuestos).
- Cada tabla tiene una clave primaria definida.
- Ejemplo: en la tabla clientes, el campo dirección se descompuso en una tabla aparte para evitar valores compuestos.

### 2FN (Segunda Forma Normal)

- Se eliminaron dependencias parciales.
- Ejemplo: en productos, la categoría no se almacena como texto repetido, sino como clave foránea hacia la tabla categorías.
- Esto evita duplicación de nombres de categorías y facilita su gestión.

### 3FN (Tercera Forma Normal)

- Se eliminaron dependencias transitivas.
- Ejemplo: en órdenes, el estado no se almacena como texto dentro de la tabla, sino como clave foránea hacia estados\_orden.
- Esto permite modificar o ampliar los estados sin alterar la estructura de la tabla principal.

## 3. Modelo de Datos en 3FN

### 1. Entidad Clientes

- id\_cliente (PK)
- nombre
- apellido
- email (único)
- telefono
- fecha\_registro
- activo (booleano)

### Relación: Direcciones

Cada cliente puede tener una o varias direcciones.

- id\_direccion (PK)
- id\_cliente (FK → Clientes)
- calle
- ciudad
- provincia
- codigo\_postal

## 2. Entidad Categorías

- id\_categoria (PK)
- nombre\_categoria (único)

## 3. Entidad Productos

- id\_producto (PK)
- nombre
- id\_categoria (FK → Categorías)
- precio
- stock
- fecha\_alta
- activo (booleano)

## 4. Entidad Estados de Orden

- id\_estado (PK)
- nombre\_estado (ej.: PENDIENTE, PAGADA, CANCELADA)

## 5. Entidad Órdenes

- id\_orden (PK)
- id\_cliente (FK → Clientes)
- fecha\_orden
- id\_estado (FK → Estados de Orden)

## 6. Entidad Detalle de Órdenes

Relaciona productos con órdenes (N:M).

- id\_detalle (PK)
- id\_orden (FK → Órdenes)
- id\_producto (FK → Productos)
- cantidad
- precio\_unitario

## 4. Justificación del Diseño

(id\_orden,fecha\_orden,estado,id\_cliente,nombre\_cliente,apellido\_cliente,email,telefono,direccion,fecha\_registro,id\_producto,nombre\_producto,categoria,precio,stock,fecha\_alta,cantidad)

d,precio\_unitario,subtotal)

#### A. Dependencias del Cliente

- id\_cliente → nombre\_cliente, apellido\_cliente, email, telefono, direccion, fecha\_registro

#### B. Dependencias del Producto

- id\_producto → nombre\_producto, categoria, precio, stock, fecha\_alta

#### C. Dependencias de la Orden

- id\_orden → id\_cliente, fecha\_orden, estado

#### D. Dependencias del Detalle

- (id\_orden, id\_producto) → cantidad, precio\_unitario, subtotal

*Clientes y Direcciones:* separación para permitir múltiples direcciones por cliente y evitar redundancia.

*Productos y Categorías:* separación para evitar repetición de nombres de categorías y facilitar su administración.

*Órdenes y Estados:* separación para que los estados sean gestionables y extensibles sin modificar la tabla principal.

*Detalle de Órdenes:* necesaria para representar la relación N:M entre órdenes y productos, evitando duplicación de datos.