

# Enunciado

---

Tenemos una tienda donde se venden una serie de productos y necesitamos almacenar datos de las facturas que se emiten. El funcionamiento principal del modelo es el siguiente.

- En cada venta se pueden vender varios productos y se emite una factura.
- De las facturas necesitamos el DNI y nombre del cliente, la fecha, el importe y el número de factura que identificará a la factura de forma única.
- Relacionado con la factura que emitimos en nuestra tienda es necesario almacenar los productos que se han comprado y se han incluido en esa factura, o sea, las líneas de factura.
- Necesitamos saber el código de cada producto, el nombre del producto, el importe y la cantidad de artículos de ese producto.

Se pide:

- Crear el modelo E/R con las entidades y atributos correspondientes.
- Identificar el atributo clave de cada entidad.
- Indicar el dominio de cada uno de los atributos.
- Identificar de qué tipo es cada uno de los atributos de las entidades (derivado, compuesto...).
- Crear las relaciones oportunas.
- Poner la cardinalidad a las relaciones.

## Identificar Entidades y Atributos

- Basándome en el enunciado, la conclusión lleva a 2 Entidades: Facturas y Productos.
- La relación será desde la tienda hacia el cliente (venta) para crear una factura.
- La disposición será: **Entidad > Atributo**
  - **Entidad FACTURA:**
    - num\_factura -> Atributo Clave (PK)
    - dni\_cliente -> Atributo
    - nombre\_cliente -> Atributo
    - fecha\_factura -> Atributo
    - importe\_total -> Atributo Derivado (de la suma de los importe de cada línea de factura)
  - **Entidad PRODUCTO:**
    - cod\_producto -> Atributo Clave (PK)
    - nombre\_producto -> Atributo
    - precio\_unitario -> Atributo
    - articulos\_por\_producto -> Atributo

## Crear Relaciones y Cardinalidades

1. **Cardinalidad: FACTURA -> almacenar:**

- **Relación almacenar:**

- num\_linea\_factura -> Atributo Clave (PK)
- cantidad\_producto -> Atributo
- importe\_linea -> Atributo Derivado (de cantidad\_producto y precio\_unitario)

- Relación **1:N** (un producto puede almacenarse mínimo en una factura y máximo en varias).
- Posteriormente, se realizará una id\_factura relacionando como **Foreign Key (FK)** la línea factura con la factura

## 2. Cardinalidad: almacenar -> PRODUCTO:

- Relación **1:N** (una factura puede almacenar 1 producto como mínimo y varios productos como máximo).
- Posteriormente, se relacionará un id\_producto como **Foreign Key (FK)** dentro de línea factura.

En el diagrama E/R, **linea\_factura** será una Entidad de relación (en forma de diamante) que conecta las entidades Factura y Producto a través de **almacenar**. En este caso, **linea\_factura no solo es una relación, sino que también actúa como una entidad que almacena información adicional sobre cada producto** en cada factura.

## Paso 3: Diagrama E/R con Cardinalidades

En el diagrama E/R:

- **FACTURA** se conecta a **almacenar** (posteriormente línea factura) con una relación de **1:N**. Esto implica que el **num\_factura** será una clave foránea en **linea\_factura** después.
- **PRODUCTO** se conecta a **linea\_factura** con una relación de **1:N**. Esto implica que el **cod\_producto** será una clave foránea en **linea\_factura** dentro de la tabla de después.

---

## Ejemplo en Texto del Diagrama E/R

Para visualizarlo de forma más estructurada:

```
Factura (num_factura, dni_cliente, nombre_cliente, fecha_factura,
importe_total)
|
| 1:N
|
Relación "Almacenar" [donde deriva linea_factura (num_linea_factura,
cantidad_producto, importe_linea) al ser una relación N:M]
|
| N:1
|
Producto (cod_producto, nombre_producto, precio_unitario)
```