**Petición**

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.

**Modelos E/R**

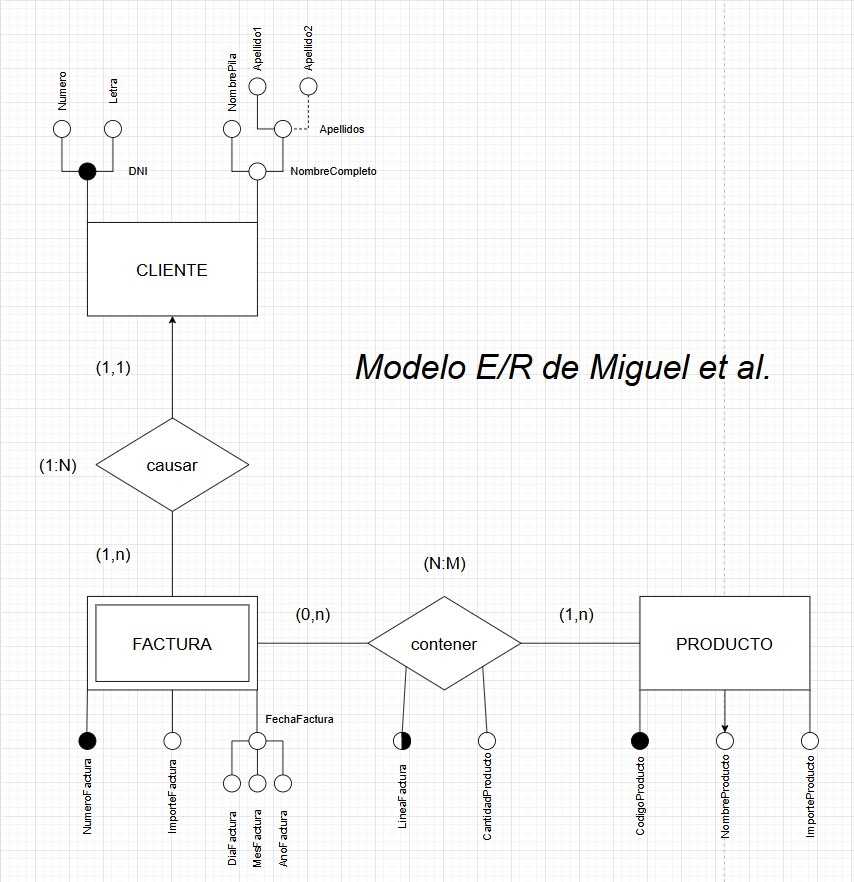
El diseño de esta base de datos tiene como objetivo el registro de las facturas emitidas por nuestra tienda. Para ello vamos a seleccionar unas entidades que nos permitan agrupar los datos que necesitamos, definiendo entidades como: *“personas, animales, objetos o cosas, que se encuentran en la vida real y son distinguibles del resto”.*

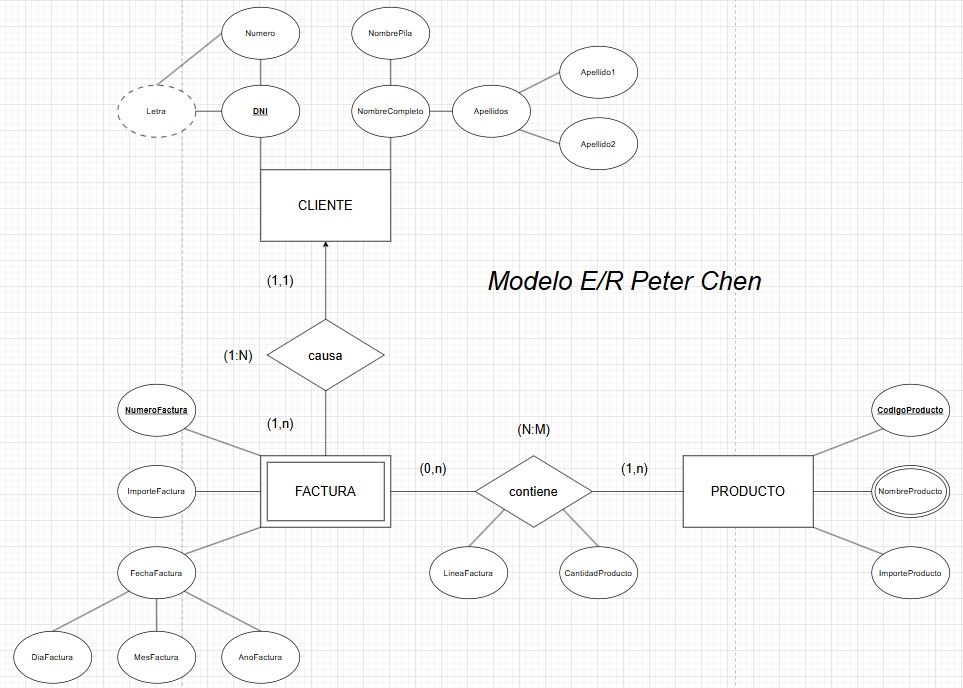
Para esta base de datos, hemos decidido establecer como entidades al cliente, a la factura y al producto. Consideramos que tener un registro de estas tres entidades es lo suficientemente importante como para considerarlas como tal. De estas tres, entendemos a las facturas como entidad débil, ya que depende tanto del cliente como del producto comprado.

Con respecto a las relaciones que estas entidades poseen entre sí, hemos distinguido las siguientes:

* La relación entre cliente y factura corresponde a causar: Que una persona se convierta en cliente es a causa de interaccionar con nosotros en forma de compra o pedido, generando de esta manera una factura. Por ello, la persona tiene que causar al menos una factura para ser considerada cliente y puede acudir a nosotros más veces causando varias facturas (1,n). A su vez, una factura no puede existir sin un cliente y solo puede ser causada por un cliente al mismo tiempo (1,1). Esto se define como relación de uno a varios (1:N) y se representa gráficamente mediante una flecha en el Modelo de de Miguel et al.
* La relación entre factura y producto se corresponde a contener. Básicamente, nosotros facturamos los productos que vendemos. Por ello, las facturas deben como mínimo contener un producto y en una venta pueden venir contenidos varios productos (1,n). Al mismo tiempo, el producto puede o no estar contenido en una factura, ya que en esta pueden venir contenidos varios productos mientras que otros no (0,n). Esta relación se define como de varios a varios (N:M).

Por tanto, sus representaciones gráficas siguiendo los modelos de de Miguel et al. y de Peter Chen serían las siguientes:





**Atributos clave de cada entidad**

Entendemos como atributo clave *“aquel atributo que permite identificar de manera única e inequívoca cada tupla de una entidad”.*

En la entidad cliente hemos elegido el DNI como atributo clave, ya que es único para cada persona y esto nos permite adaptarlo como identificador único de nuestros clientes, ahorrándonos asignar uno propio.

Para la entidad factura tenemos el número de factura como atributo único e irrepetible de cada factura guardada en la tabla.

De manera similar, en producto asignamos el código del producto como distintivo de cada producto. De esta manera, aunque vendamos de un catálogo de productos limitado (NombreProducto lo consideramos multivaluado por esta razón), cada uno tiene un distintivo y localizador único.

Cabe destacar el atributo “LineaFactura” de la tabla intermedia “contener” como posible atributo clave. Con este atributo identificamos las líneas de factura de cada factura. Es posible que, en cada factura, las líneas de factura empiecen desde 1 a n, pero un método que ayuda a su mejor localización es considerar todas las líneas de factura como únicas. Por ejemplo, la factura 1 tendría las líneas de factura de la 1 a la 5, la factura 2 comenzaría con la línea de factura 6, etc. De esta manera podemos combinar las tablas con un identificador de línea único.

**Características de los atributos**

Por último, desglosamos las características de los atributos de las entidades y relaciones. Destacamos los tipos de atributo, el formato del dato guardado, el dominio o rango del dato introducido y un ejemplo de cada uno para hacerlo más claro. En este caso ejemplificaremos con un concesionario. La lista es la siguiente:

*Entidades:*

* CLIENTE:
  + DNI

Tipo atributo: Clave Primaria Compuesta (PK – Primary Key)

Formato de dato: Texto (String)

Dominio: 00000000A – 99999999Z

Ejemplo: 45454545J

Solo su composición se considera PK. Está compuesto de:

* + - Numero

Tipo atributo: Simple Univaluado

Formato de dato: Entero (Integer)

Dominio: 00000000 - 99999999

Ejemplo: 45454545

* + - Letra

Tipo atributo: Simple Univaluado Derivado

Formato de dato: Alfanumérico (Alphanumerical)

Solo letras y por claridad de los datos solo mayúsculas.

Dominio: A – Z

Ejemplo: J

El número del DNI genera la letra, por lo tanto, es derivado. Reflejamos esta relación en el diagrama de Peter Chen ya que este nos permite representar que es derivado mediante la línea de puntos. En el diagrama de de Miguel et al. lo dejamos sin la unión para evitar confusión ya que su representación es la misma.

* + NombreCompleto

Tipo atributo: Compuesto Univaluado

Formato de dato: Texto (String)

Dominio: 1 a 80 caracteres

Ejemplo: Adrián Olmedo de David

Somos bastante holgados con los caracteres que permitimos por si acaso.

* + - NombrePila

Tipo atributo: Simple Univaluado

Formato de dato: Texto (String)

Dominio: 1 a 20 caracteres

Ejemplo: Adrián

* + - Apellidos

Tipo atributo: Compuesto Univaluado

Formato de dato: Texto (String)

Dominio: 1 a 60 caracteres

Ejemplo: Olmedo de David

* + - * Apellido1

Tipo atributo: Simple Univaluado

Formato de dato: Texto (String)

Dominio: 1 a 30 caracteres

Ejemplo: Olmedo

* + - * Apellido2

Tipo atributo: Simple Univaluado Opcional

Formato de dato: Texto (String)

Dominio: 1 a 30 caracteres

Ejemplo: de David

El segundo apellido es opcional, ya que podemos tratar con clientes que carezcan de segundo apellido.

* FACTURA:
  + NumeroFactura

Tipo atributo: Clave Primaria (PK – Primary Key)

Formato de dato: Entero (Integer)

Dominio: 1-999999

Ejemplo: 003621

* + ImporteFactura

Tipo atributo: Simple Univaluado

Formato de dato: Decimal (Decimal)

Dominio: 1000 - 999999

Ejemplo: 34995,59

Considerando nuestros productos, este rango debería ser suficiente.

* + FechaFactura

Tipo atributo: Compuesto Univaluado

Formato de dato: Temporal (Date)

Dominio: 01/01/1900 – 31/12/2999

Ejemplo: 11/11/2024

Los días y meses podrían considerarse multivaluados si se desea, ya que su rango es suficientemente corto.

* + - DíaFactura

Tipo atributo: Simple Univaluado

Formato de dato: Entero (Integer)

Dominio: 01-31

Ejemplo: 11

* + - MesFactura

Tipo atributo: Simple Univaluado

Formato de dato: Entero (Integer)

Dominio: 01-12

Ejemplo: 11

* + - AnoFactura

Tipo atributo: Simple Univaluado

Formato de dato: Entero (Integer)

Dominio: 1900 - 2999

Ejemplo: 2024

* PRODUCTO:
  + CodigoProducto

Tipo atributo: Clave Primaria (PK – Primary Key)

Formato de dato: Texto (String)

Dominio: 10 a 10 caracteres

Ejemplo: 12dc300y97

Queremos que la variación de códigos sea la mayor posible.

* + NombreProducto

Tipo atributo: Simple Multivaluado

Formato de dato: Texto (String)

Dominio: Mazda CX-30, Mazda CX-5, Mazda CX-60, Mazda CX-80, Mazda MX-30, Mazda2, Mazda2 Hybrid, Mazda3 y Mazda MX-5

Ejemplo: Mazda MX-5

* + ImporteProducto

Tipo atributo: Simple Univaluado

Formato de dato: Decimal (Decimal)

Dominio: 1000 - 999999

Ejemplo: 34995,59

Considerando nuestros productos, este rango debería ser suficiente.

*Relaciones:*

* Contener:
  + LineaFactura

Tipo atributo: Clave Alternativa (Unique)

Formato de dato: Entero (Integer)

Dominio: 1-999999

Ejemplo: 049642

En este caso y como comentabamos anteriormente, estamos considerando cada línea de factura única, por ello este rango.

* + CantidadProducto

Tipo atributo: Simple Univaluado

Formato de dato: Entero (Integer)

Dominio: 0 - 999

Ejemplo: 40

Considerando nuestros productos, este rango debería ser suficiente.

**Conclusión**

Con este diseño de base de datos podemos llevar un registro de las facturas que emitimos, de los datos de los clientes en caso de ser necesarios, y actúa también como un catálogo de los productos de ofertamos.