# Requisitos V2

1. En la carpeta modelos se han de crear los siguientes clases:

**Factura**

- identificador-> int
- codigoFactura -> String
- importeBase-> double
- descuento-> double
- iva-> double
- totalAPagar -> double
- fechaEmision-> LocalDate
- fechaVencimiento-> LocalDate
- pagada -> boolean
- lineaFactura-> List<LineaFactura>
- cliente -> Cliente

**LineaFactura**

- identificador-> int
- factura-> Factura
- producto-> Producto
- cantidad-> int

Todos los modelos han de contar con los atributos privados que se detallan , al igual que el constructor vacío, copia y completo, los métodos get y set de todos los atributos y los métodos toString, equals y hashcode.

2. Se añade a la clase Producto un nuevo atributo:
   precio -> double
   Hay que añadir los métodos getters y setters de este nuevo atributo e indicarlo en constructores, toString , equals, etc.
3. En el paquete src>java>utilidades se ha de crear las siguientes clases:
   a. UtilidadesFactura.class
   b. UtilidadesLineaFactura.class

4. En la clase UtilidadesFactura hay que crear los siguientes métodos:

   a. public boolean esFacturaVencida(Factura factura).

      Que devuelve true si la fecha de vencimiento de la factura que se le pasa es mayor o igual que la fecha actual y false en caso contrario.

   b. public double calcularBaseFactura(Factura factura)

      Que devuelve el importe base que debe tener la factura que se le pasa. Para ello recorre todas las Linea Factura de la factura que se le pasa y va acumulando en una variable, la suma del importe del Producto de la LineaFactura multiplicada por la cantidad de dicho producto, también indicada en la linea.

   c. public double calcularTotalAPagar(Factura factura)

      Que devuelve el totalAPagar que debe tener la factura que se le pasa. Para ello hace la siguiente operación:
      (importeBase - descuentos) * iva
      Todos estos datos obtenidos de la factura que se le pasa.

.

5. En la clase UtilidadesProducto hay que crear los siguientes métodos:

    a. public List<Producto> getPorTipo(List<Producto> productos, TipoProducto tipo).

    El método devuelve la lista de productos del tipo que se le pasa como parámetro.

    b. public List<Productos> getPorAlmacen(List<Producto> productos, Almacen almacen)

    El método devuelve la lista de productos del almacen que se le pasa como parámetro.

6. En la clase UtilidadesCliente hay que crear el siguiente método:

    a. public boolean esDniValido(Cliente cliente).

    El método devuelve true si el dni del cliente que se le pasa, tiene 9 caracteres y de los cuáles los 8 primeros son números y el último una letra, y devuelve false en caso contrario.

# *Proyecto Java - Requisitos - V2*

***Profesor: Luis Javier López López***