

1.- Diseña una jerarquía de clases para los diferentes productos que se pueden encontrar en un supermercado (Alimentación, Droguería y Textil). Todos tienen un conjunto de características comunes (código, nombre y precio) y una serie de características específicas de cada producto. El código que crea los objetos de cada tipo se llamará CreacionObjetos. [comprobar si existe ese producto en el clase Supermercado](#)

Los productos de alimentación tienen fecha de caducidad (no se podrá introducir un producto si la fecha de caducidad no es al menos de 2 días, en ese caso lanza la excepción propia **FechaNoValidaException** con el mensaje apropiado), los de droguería tienen marca y los de textil color.

La fecha será introducida por el usuario pidiéndole día, mes y año por separado y se convertirá a formato LocalDate. En caso de no ser una fecha válida, el programa mostrará una excepción denominada **FechaNoValidaException** con el mensaje apropiado.

El código debe adaptarse a un tamaño de 5 caracteres de los cuales el primero y el último son letras en mayúscula y el resto números. Utiliza expresiones regulares para controlarlo, lanzando una excepción **CodigoInvalidoException** que crearás si no cumple la notación.

Debes crear métodos que lean los valores que se piden al usuario y, en el caso de valores numéricos, que controlen que el valor introducido es numérico y permitan repetir el intento las veces que sea necesario. Esto lo harás en la clase Lector y cada método recibirá el mensaje que se muestra al usuario y devolverá el valor leído si es correcto.

Los métodos que comprueben que el código es correcto, lo mismo que los que comprueban que la fecha de caducidad es correcta y bien formada, deben estar en una clase denominada Validaciones.

Crea un carrito de la compra en el que se pueden incluir productos y emitir un ticket en el que figuren los datos de cada producto del carrito, incluyendo su precio y el importe total de la compra. Si se quiere más de una unidad de un producto, se pide varias veces, para simplificar.

Para llevar a cabo lo explicado, el programa al arrancar mostrará el menú:

## PROCESO DE CREACIÓN DEL ALMACÉN

1. Introducir producto alimenticio
2. Introducir producto textil
3. Introducir producto droguería
4. Terminar.

Elija opción:

Con ese menú, se creará un ArrayList con varios productos para poder simular los productos de un supermercado y poder hacer compras. Ese ArrayList estará en la clase Supermercado y solo se usará en dicha clase.

Entonces empezará el proceso de compras, mostrando para cada cliente el listado de productos y pidiendo al cliente que seleccione el producto que quiere comprar por código, hasta que ya no quiera más productos. En ese momento, se procede a imprimir el ticket con los distintos productos y el precio total a pagar.

Para cada cliente nuevo se visualizará lo siguiente:

**;;;Acaba de entrar en el supermercado virtual!!!!**

**Lista de productos**

**1 Nombre:peras Precio:2.0 fecha caducidad alimento:23/2/1989**

**2 Nombre:toalla Precio:8.0 Color:azul**

**3 Nombre:gel Precio:2.0 Marca:Legrain**

**Desea comprar producto?(S/N)**

Irá creando un carrito con los productos que el cliente va comprando. Cuando el cliente termine (responde N), se calcula el importe de su compra y se empieza con un nuevo cliente (previamente se pregunta si hay un nuevo cliente)

Cuando no haya más clientes, terminaremos la ejecución del programa.

Esquema del proyecto:

- └─ Supermercado
  - └─ src
    - └─ productos
      - > Alimento.java
      - > Drogueria.java
      - > Producto.java
      - > Textil.java
    - └─ supermercado
      - > Carrito.java
      - > MenuApp.java
      - > Supermercado.java
    - └─ utilidades
      - > CreacionObjetos.java
      - > Lector.java
      - > Validaciones.java
    - > JRE System Library [JavaSE-1.8]

Proyecto donde se crean las excepciones:

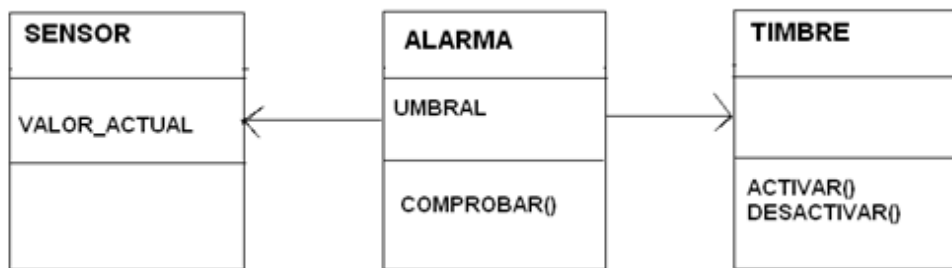
- └─ Excepciones
  - > JRE System Library [JavaSE-14]
  - └─ src
    - └─ excepciones
      - > CodigoIncorrectoException.java
      - > FechaNoValidaExcepcion.java

### Casos de Prueba:

- 1.- Todo va bien. Código y fecha de caducidad (correcta en formato y tiempo)
- 2.- La fecha que se introduce para la caducidad es correcta en formato, pero no tiene al menos dos días de caducidad. Se muestra el mensaje y no se vuelve a pedir. El producto no se crea
- 3.- La fecha que se introduce para la caducidad es incorrecta en formato. Se vuelve a pedir.

Primero comprobamos si es correcta en formato y si lo es se procede a comprobar si el producto ya está caducado.

- 2.- Crea una clase denominada *Alarma* cuyos objetos activen un objeto de tipo *Timbre* cuando el valor medido por un sensor supere un umbral preestablecido:



Implementa en Java todo el código necesario para el funcionamiento de la alarma, suponiendo que la alarma comprueba si debe activar o desactivar el timbre cuando se invoca el método *comprobar()*.

Crea una subclase de Alarma denominada *Alarma\_Luminosa*, que además de activar el timbre, encienda una luz (que representaremos con un objeto de tipo *Bombilla*). Aseguraos de que cuando se activa la alarma luminosa se enciende la luz de alarma y también suena la señal sonora asociada al timbre.