

Examen 2 Programación	
(Objetos, getters y setters, toString, arrays (uni y bidi), funciones, try..catch)	
Nombre del alumno/a:	Calificación

Creación de la clase Producto

La profesora ha creado, en el proyecto entregado, una clase llamada Producto. Este producto tendrá los siguientes campos:

- **nombre** del producto
- **fecha** de compra, fecha en la que se compró este producto
- **número de Albarán**, tendrá la siguiente forma "01092002TOR01"
- **pvp**, precio por unidad
- **unidades** vendidas
- **preciototal** --> saco el precio total, según el pvp, el nº de unidades y el dto de la tienda y el mes indicado en la fecha
- Un array bidimensional llamado **descuentos** que me dice, según el nº de tienda (de 15) y el nº de mes, qué descuento se aplica (aleatorio entre 0%, 10%, 20%, 35%, 50%)

El objetivo será completar la clase Producto para que todas las funciones del main trabajen correctamente.

1._ (1,5 puntos) Función pideTienda

Para que el constructor de Producto funcione correctamente hemos de pasarle un número de tienda. Éste es el número que se pedirá que introduzca al usuario. Este número tendrá que:

- Ser un número entero. Si introduzco algo que no es un número, debe poner ".... Algo ha ido mal" y **VOLVER A PEDÍRMELO**.
- Estar entre 0 y 14, puesto que existen 15 tiendas. Si no es así, se volverá a pedir el número

Ejemplo:

```

-----
Tienda: ertwert
...algo ha ido mal
Tienda: -1
Tienda: 14|

```

PUNTUACIÓN	
Firma de la función	0.4 ptos
Bucle correcto	0.4 ptos
Control errores correcto	0.7 ptos

2._ (1 punto) Función setFecha

A esta función le llegará un String con el siguiente formato "yyyy-mm-dd". Nosotros tenemos que encargarnos de convertir ese String al tipo LocalDate del campo fecha de la clase producto, CONTROLANDO todos los errores. Si algo sale mal, la función debe poner un null en este campo fecha.

PUNTUACIÓN	
Firma función correcta	0.1 ptos
Rellena la fecha correctamente	0.9 ptos
Pone la fecha "a pelo"	0.1 ptos

3._ (1 puntos) Función getFecha

Normalmente esta función devolvería un Localdate, pero nosotros vamos a modificarla para que devuelva un String, y así, poder mostrar la fecha guardada. El formato en que debe mostrarse la fecha será el que se refleja en la imagen (tres letras para el día, un espacio y luego la fecha formateada en español). He aquí un ejemplo

```
Fecha (formato yyyy-mm-dd): 2323-12-01
Tienda: 0
Fecha: sáb 01-12-2323
BUILD SUCCESSFUL (total time: 28 seconds)
```

Tened cuidado porque si con el setFecha algo ha salido mal, la fecha será null y yo debo devolver un null.

```
Fecha (formato yyyy-mm-dd): 1212-2-3
Tienda: 6
Fecha: null
```

PUNTUACIÓN	
Firma función correcta	0.3 pts
Devuelve fecha formato correcto	0.6 pts
Devuelve fecha en otro formato	0.1 pts
Devuelve null si algo va mal	0.1 pts

4._ (2 puntos) Función setAlbaran

Esta función se encargará de rellenar el campo albaran de la clase Producto. El albarán será un String de la siguiente forma:

- Si la fecha es correcta: "12122022TOR01". Es decir, el día con dos dígitos, el mes con dos dígitos, el año con 4 dígitos, seguido de las tres primeras letras del nombre en mayúscula y luego el número de tienda con dos dígitos.
- Si la fecha no es correcta: "DDMMYYYYTOR01"

Esta función podrá hacer uso de la función getFecha realizada anteriormente.

Ejemplos:

	Nombre del producto: fresadora
Fecha (formato yyyy-mm-dd): 2222-02-02	Fecha (formato yyyy-mm-dd):
Tienda: 6	Tienda:
Fecha: sáb 02-02-2222	9
Albarán: 02022222TOR06	Fecha: null
	Albarán: DDMMYYYYFRE09

PUNTUACIÓN	
Pone el albarán correctamente	2 pts
Rellena el albarán "a mano" o pone otro tipo de albarán	0.5 pts

5._ (2 puntos) Función setDescuentos

Esta función se encargará de rellenar el array bidimensional de descuentos. Este array tendrá tantas filas como tiendas haya y tantas columnas como meses hay. En cada casilla se generará un número aleatorio entre 0 y 60, y si este número está:

- Entre 0 y 10 (excluido): se le asignará un descuento del 0%
- Entre 10 y 20 (excluido): se le asignará un descuento del 10%
- Entre 20 y 35 (excluido): se le asignará un descuento del 20%
- Entre 35 y 50 (excluido): se le asignará un descuento del 35%
- Al resto se le asignará un descuento del 50%

La generación de este número se realizará mediante una función llamada generaDescuento.

Un pequeño ejemplo de lo que se mostrará:

```
Tienda 0:
20 20 10 20 20 50 20 20 35 20 10 35
Tienda 1:
0 50 10 35 10 20 20 50 10 20 20 35
Tienda 2:
10 20 0 35 20 10 35 35 50 35 20 0
Tienda 3:
50 0 10 0 0 0 20 20 20 0 35 50
Tienda 4:
50 20 20 35 10 20 20 50 35 50 10 35
```

PUNTUACIÓN	
Rellenado correcto del array mediante la función generaDescuento	2 ptos
Rellenado correcto del array SIN usar funciones	1 pto
Genera números erróneos	0.5 pto
Rellena el array "a mano"	0.5 pto
Si hay algún error "rojo"	0 ptos

6._ (2,5 puntos) Función setPrecioTotal

Esta función se encargará de rellenar el atributo precioTotal de la siguiente forma. El precio total irá en función del pvp y el número de unidades. A este precio se le quitará el descuento indicado en la tabla descuentos según la tienda donde estemos y el mes que contenga la fecha. Es decir, para obtener el descuento tendremos que crear una función llamada **obtenerMes**, que me devuelva el mes contenido en la fecha. Si la fecha es mala o no puedo realizar algún cálculo, esta función devolverá -1. Una vez obtenido el mes, iremos a la tabla descuentos y buscaremos en la tienda almacenada y el mes obtenido, para recuperar el % de descuento y aplicárselo al precio. Si la función obtenerMes devuelve -1, entonces supondremos que, por defecto, el mes será el 0. Cuidado con las fechas nulas!!

Ejemplo de salidas:

```
20 35 20 30 0 0 35 10 20 35 20 30
Tienda 9:
20 20 50 0 0 50 10 0 10 20 10 20
Tienda 10:
35 10 35 20 10 0 20 20 50 35 50 50
Tienda 11:
0 20 50 20 50 10 35 10 35 50 35 50
Tienda 12:
35 0 50 0 20 50 20 20 35 35 10 20
Tienda 13:
0 0 10 10 10 50 20 0 20 50 35 50
Tienda 14:
35 35 0 20 50 10 20 10 50 35 20 10
14.4
Producto{Nombre=Fresadora, fecha=2022-12-25, albaran=25122022FRE09, pvp=0.6, unidades=30, tienda =9, precioTotal=14.4}
BUILD SUCCESSFUL (total time: 20 seconds)
```

Con fecha null:

Tienda 14:

10 0 50 10 50 50 20 20 50 35 35 10

16.2

Producto{Nombre=Tornillo, fecha=null, albaran=DDMMYYYYTOR14, pvp=0.6, unidades=30, tienda =14, precioTotal=16.2}

funcion obtenerMes (0, 1, 11, 22, 33, 44, 55, 66, 77, 88, 99)

PUNTUACIÓN	
Rellenado correcto del precio con el descuento y la función obtenerMes	2,5 pts
Rellenado correcto del precio con el descuento pero sin la función	1,5 pto
Rellenado del precio sin descuento	0,5 pts
Si hay algún error “rojo”	0 pts

Criterios de evaluación del examen (en cada ejercicio se tocan todos)			
RA1.	CE a, b, c, d, e, f, g, h, i	RA3.	CE a, b, c, d, e, f, g

Para cada ejercicio se seguirá la siguiente rúbrica:

Porcentaje	Comentario
100%	El programa/clase/función implementado se ajusta perfectamente a las especificaciones/requerimientos. La ejecución es correcta para todas las circunstancias. El código es legible (debidamente comentado, estructurado, modular y con nombres de variables/funciones/clases significativos)
75%	El programa/clase/función implementado se ajusta perfectamente las especificaciones/requerimientos. La ejecución es correcta para todas las circunstancias. El código no es lo suficientemente legible (comentarios, estructura, nombres, etc.)
50%	El programa/clase/función implementado se ajusta mayoritariamente las especificaciones/requerimientos y el código es legible, pero en algunos casos no funciona correctamente o tiene algún error menor.
25%	El programa/clase/función implementado no se ajusta totalmente las especificaciones/requerimientos, el código no es legible y no funciona para la mayoría de los casos.
0%	El programa/clase/función implementado no se ajusta a las especificaciones, no resuelve el problema de forma algorítmica o atenta contra la programación estructurada (saltos indiscriminados, uso de break, continue o goto)

Una vez terminado, el código deberá subirse a moodle en la actividad creada a tal efecto