

ACTIVIDAD 2 - SEGUNDA EVALUACIÓN

Trabajas en la compañía **AvellanedaSoft S.L** y tu jefa de proyecto, @profeMelola, te pide que **mejores un programa** que gestiona personal de un instituto y productos de su cafetería.

Se te proporciona el programa. Descárgalo, monta el proyecto y podrás ejecutarlo sin problema. Seguro que te suena el código fuente.

LAS MEJORAS A REALIZAR SON:

1. USO DE LA INTERFACE **Validable** y su método abstracto [3.5 puntos]

[0,5 puntos] Debe desaparecer el método abstracto `public abstract boolean validar();` tanto de la clase **Persona** como de la clase **Producto**.

En el paquete **es.daw.poo.model** hay una interface llamada **Validable**.

Debes usarla de forma que todos los objetos del programa (tanto personas como productos) implementen su método:

```
public boolean validar() throws NotValidException;
```

Como puedes observar, dicho método propaga la excepción **NotValidException**.

[3 puntos: 0,5 puntos por cada excepción bien propagada con su mensaje correspondiente]

Crea, lanza y propaga la excepción correctamente en los diferentes métodos “validar” (tendrás que hacerlo 6 veces)

Dependiendo de donde se realice la excepción, el mensaje de la misma será diferente:

- **ProductoCongelado** ⇒ “La temperatura de congelación es superior a cero”
- **ProductoFresco** ⇒ “El país no está permitido”
- **ProductoRefrigerado** ⇒ “El código de supervisión no cumple con el patrón”
- **Profesor** ⇒ “El profesor tiene menos de dos años de experiencia”
- **Estudiante** ⇒ “El NIA no cumple con el patrón”
- **EstudianteInternacional** ⇒ “El país no está permitido”

2. EXCEPCIÓN **NotValidException** [0,5 puntos]

Debes crear dicha excepción en el paquete **es.daw.poo.exceptions**.

3. CAPTURAR LA EXCEPCIÓN AL MOSTRAR LOS PRODUCTOS Y PERSONAS

[1 punto]

En la clase principal debes añadir la gestión de la excepción a la hora de mostrar:

- Los productos no válidos. [0,5 puntos]

```
* Productos no válidos:
La temperatura de congelación es superior a cero
Producto [fechaCaducidad=2025-02-13, numeroLote=5]-> ProductoCongelado [temperaturaCongelacion=5]

El país no está permitido
Producto [fechaCaducidad=2024-02-16, numeroLote=2]-> ProductoFresco [paisOrigen=Marruecos]

El código de supervisión no cumple con el patrón
Producto [fechaCaducidad=2024-03-13, numeroLote=4]-> ProductoRefrigerado [codigoSupervision=1234]
```

- El personal no válido. [0,5 puntos]

```
* Personal que no ha pasado la validación:
El profesor tiene menos de dos años de experiencia
Persona [nombreCompleto=Mari Luz, direccion=Mari Luz]-> Profesor [aniosExperiencia=1]

El país no está permitido
Persona [nombreCompleto=Mark Zuckerberg, direccion=Donde vive Mark]-> Estudiante [NIA=12345678C]->EstudianteInternacional [paisOrigen=Marruecos]

El NIA no cumple con el patrón
Persona [nombreCompleto=Scarlett Johansson, direccion=Donde vive Scarlett]-> Estudiante [NIA=A234]
```

4. MÉTODOS NATURALES DE ORDENACIÓN [3 puntos]

[1 punto] El personal del centro será ordenado de forma natural por su nombre completo.

[1 punto] Los productos serán ordenados de forma natural por su fecha de caducidad.

[1 punto] En la clase principal ordena las diferentes colecciones y muestra su resultado por consola:

```
PERSONAS ORDENADAS POR NOMBRE COMPLETO ASCENDENTE:
Persona [nombreCompleto=Chema, direccion=Chema]-> Profesor [aniosExperiencia=20]
Persona [nombreCompleto=Elon Musk, direccion=Donde vive Elon]-> Estudiante [NIA=12345678A]->EstudianteInternacional [paisOrigen=España]
Persona [nombreCompleto=Luis Maeso, direccion=Donde vive Luis]-> Estudiante [NIA=12345678B]
Persona [nombreCompleto=Mari Luz, direccion=Mari Luz]-> Profesor [aniosExperiencia=1]
Persona [nombreCompleto=Mark Zuckerberg, direccion=Donde vive Mark]-> Estudiante [NIA=12345678C]->EstudianteInternacional [paisOrigen=Marruecos]
Persona [nombreCompleto=Scarlett Johansson, direccion=Donde vive Scarlett]-> Estudiante [NIA=A234]

PRODUCTOS ORDENADOS POR FECHA ASCENDENTE:
Producto [fechaCaducidad=2024-02-16, numeroLote=6]-> ProductoFresco [paisOrigen=España]
Producto [fechaCaducidad=2024-02-16, numeroLote=2]-> ProductoFresco [paisOrigen=Marruecos]
Producto [fechaCaducidad=2024-03-13, numeroLote=1]-> ProductoRefrigerado [codigoSupervision=A123]
Producto [fechaCaducidad=2024-03-13, numeroLote=4]-> ProductoRefrigerado [codigoSupervision=1234]
Producto [fechaCaducidad=2025-02-13, numeroLote=5]-> ProductoCongelado [temperaturaCongelacion=5]
Producto [fechaCaducidad=2025-02-14, numeroLote=3]-> ProductoCongelado [temperaturaCongelacion=-20]
```

5. INTEGRAR EN UN ÚNICO DAO [2 puntos]

Como tanto la clase Persona, como la clase Producto son Validable, mediante el polimorfismo podríamos juntar la creación de sus objetos en una única clase DAO llamada **ValidableDAO**.

- Crea dicha clase.[1 punto]
- En la clase principal obtén todos los objetos validables y píntalos por consola. [1 punto]