

## **Actividad evaluable 2ª EVALUACIÓN** **UD5: Programación modular**

### **ACTIVIDAD eval02.java**

Realiza en Java una divertidísima aplicación que simule la gestión de alimentos de una nevera.  
El programa *main* seguirá los siguientes pasos:

- 1- Pedirá el número de serie del aparato que validará en bucle llamando a la rutina

```
public static boolean validaNumeroSerie(String num)
```

- 2- Pedirá la capacidad de la nevera que validará en bucle llamando a la rutina:

```
public static boolean validaCapacidad(int cap)
```

- 3- Creará un vector que tendrá la capacidad validada en el paso 2. Este vector será la nevera y lo rellenaremos con “N”.
- 4- Creará un vector que tendrá la mitad de la capacidad validada en el paso 2. Este vector será el congelador y lo rellenaremos con “C”.

- 5- Imprimirá por pantalla la nevera y el congelador uno encima del otro llamando a la rutina:

```
public static void imprimirPantalla(String[] n, String[] c)
```

- 6- Realizará la compra para guardarla en la nevera llamando a la rutina:

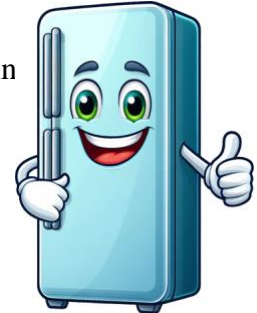
```
public static void hacerCompra(String[] n)
```

- 7- Repetimos el paso 5

- 8- Pasaremos la compra de la nevera al congelador (la que quepa) llamando a la rutina:

```
public static void congelarCompra(String[] n, String[] c)
```

- 9- Repetimos el paso 5



**RUTINA 1:** `public static boolean validaNumeroSerie(String num)`

-*num* tendrá contendrá una cantidad de dígitos impar, siendo esta cantidad mayor o igual a 3.

-la primera mitad del *num* será un número par y la segunda mitad un número impar. P ej:

123 : 1 es IMPAR (ERR) y 3 es IMPAR (OK). *num* no es correcto.

10813: 10 es PAR (OK) y 13 es IMPAR (OK). *num* es correcto.

**RUTINA 2:** `public static boolean validaCapacidad(int cap)`

-*cap* tendrá que ser un número par mayor o igual que 4.

**RUTINA 3:** `public static void imprimirPantalla(String[] n, String[] c)`

-Imprime todos los elementos del vector *n* y luego todos los elementos del vector *c*.

**RUTINA 4:** `public static void hacerCompra(String[] n)`

Se solicitará al usuario una lista de alimentos que irán separados por guión y que acabará en “-fin”. Si la lista no supera la validación, se volverá a solicitar en bucle. P ej:

manzanas-peras-filete de delfin (ERR)

manzanas-peras-filete de delfin-fin (OK)

Si validará también que la lista de alimentos cabe en la nevera. Es decir, si la lista anterior es de 5 alimentos, pero en nuestra nevera solo caben 4, se volverá a solicitar.

Una vez superada la validación anterior, se moverán los elementos de la lista a la nevera.

**RUTINA 5:** `public static void congelarCompra(String[] n, String[] c)`

Se moverán todos los alimentos desde el vector *n* hasta completar el vector *c*.

Cada vez que se mueva un alimento desde la nevera hasta el congelador, el alimento dejará de aparecer en el vector *n*, su espacio se sustituirá por una “N” y aparecerá en el vector *c*.

Un ejemplo de ejecución sería el siguiente:

**PASO1: NUMERO DE SERIE**

Introduzca numero de serie de la nevera:

12

ERROR! La longitud de la cadena es menor a 3!

Introduzca numero de serie de la nevera:

123

ERROR! La primera mitad del numero NO ES PAR!

Introduzca numero de serie de la nevera:

10813

**PASO2: CREACION DE LA NEVERA Y EL CONGELADOR**

Introduzca la capacidad de la nevera:

3

ERROR! La capacidad es impar!

ERROR! La capacidad ha de ser minimo 4 litros!

Introduzca la capacidad de la nevera:

4

Nevera y congelador creados!:

-----

N

N

N

N

-----

C

C

-----

**PASO3: HACER LA COMPRA**

Introduzca los alimentos separados por un guion y acabado en 'fin':

manzanas-peras-aguacate-salmon-filete

ERROR! La lista de alimentos no acaban en -fin!

Introduzca los alimentos separados por un guion y acabado en 'fin':

manzanas-peras-aguacate-salmon-filete-fin

ERROR! Intentas poner 5 alimentos y solo caben 4!

Introduzca los alimentos separados por un guion y acabado en 'fin':

manzanas-peras-aguacate-salmon-fin

Después de hacer la compra:

-----

manzanas

peras

aguacate

salmon

-----

C

C

-----

**PASO4: CONGELAR LA COMPRA**

Después de congelar la compra:

-----

N

N

aguacate

salmon

-----

manzanas

peras

-----

## Rúbrica de evaluación

Hitos	Puntuación
1-El programa principal tiene la estructura solicitada	1
2-Rutina <code>validaNumeroSerie</code>	2
3-Rutina <code>validaCapacidad</code>	1
4-Rutina <code>imprimirPantalla</code>	0,5
5-Rutina <code>hacerCompra</code>	3
6-Rutina <code>congelarCompra</code>	2
7-Código bien estructurado y comentado	0,5