

Ejercicios Programación funcional

[Ejercicio 1](#)[Ejercicio 2](#)[Ejercicio 3](#)[Ejercicio 4](#)

Hacer 3 versiones en todos los ejercicios:

1. Con clases anónimas inline.
2. Con una expresión lambda.
3. Con referencias a métodos (cuando se pueda).

Ejercicio 1

Aplica programación funcional a todos los ejercicios de colecciones.

Ejercicio 2

Crea una clase que tenga el siguiente atributo: `int[] array = { 1, 2, 3, 4, 5, 6, 7, 8 };`

Realiza los siguientes métodos:

- `public void showArray(Consumer<Integer> consumer)`: para mostrar los elementos del array tal y como indique el *consumer*.
- `public void showArray(BiConsumer<Integer, Integer> biconsumer)`: para mostrar los índices y los elementos del array tal y como indique el *biconsumer*.
- `public int countArray(Predicate<Integer> predicate)`: para contar qué elementos cumplen cierta condición.
- `public int[] filterArray(Predicate<Integer> predicate)`: para filtrar el array por cierta condición.
- `public int[] transformArray(UnaryOperator<Integer> unaryOper)`: para transformar el array en otro array en función de un cálculo.
- Método que obtenga la siguiente salida por consola utilizando los métodos anteriores:

```
showArray para mostrar los elementos en la misma línea: 1 2 3 4 5 6 7 8
showArray para mostrar los elementos cada uno en una línea:
1
2
3
4
5
6
7
8
showArray para mostrar los elementos con su índice:
Elemento 0: 1
Elemento 1: 2
Elemento 2: 3
```

```
Elemento 3: 4
Elemento 4: 5
Elemento 5: 6
Elemento 6: 7
Elemento 7: 8
```

```
countArray para contar pares: 4
countArray para contar impares: 4
countArray para contar los mayores de 3: 5
```

```
filterArray para filtrar pares: [2, 4, 6, 8]
filterArray para filtrar impares: [1, 3, 5, 7]
filterArray para filtrar los mayores de 3: [4, 5, 6, 7, 8]
```

```
transformArray para obtener el doble: [2, 4, 6, 8, 10, 12, 14, 16]
transformArray para obtener el triple: [3, 6, 9, 12, 15, 18, 21, 24]
```

Ejercicio 3

Realiza el *Ejercicio2* pero con listas en lugar de arrays.

Ejercicio 4

Realiza el siguiente mapa:

- Clave: nombre de empleados
- Valor: sueldo de empleados

Para dicho mapa, realiza lo siguiente:

- Muestra los datos del mapa utilizando el método `forEach`:

Empleado: Ana	Sueldo: 2000,96
Empleado: Rodrigo	Sueldo: 2500,74
Empleado: Juan	Sueldo: 1000,45
Empleado: Antonio	Sueldo: 3000,32
Empleado: Pepe	Sueldo: 1500,58

- Utilizando el método `computeIfAbsent`, introduce en el mapa a "Luis" con un sueldo base de 1000 euros:

Empleado: Luis	Sueldo: 1000,00
----------------	-----------------

- Utilizando el método `computeIfPresent`, súbele el sueldo a "Luis" un 20%:

Empleado: Luis	Sueldo: 1200,00
----------------	-----------------

- Utilizando el método `replaceAll`, súbele el sueldo a todos un 10%:

Empleado: Ana	Sueldo: 2201,06
Empleado: Luis	Sueldo: 1320,00
Empleado: Rodrigo	Sueldo: 2750,81
Empleado: Juan	Sueldo: 1100,50
Empleado: Antonio	Sueldo: 3300,35
Empleado: Pepe	Sueldo: 1650,64