

## Ejercicios de clase de Scala programación funcional

Utilizando el lenguaje de programación scala, resuelva los siguientes apartados:

1. Escribir una función llamada *nElemento* que devuelva el elemento *n* de una lista. El primer elemento de la lista se corresponde con  $n = 1$

Como ejemplo:

```
scala > nElemento(3, List ("enero", "febrero", "marzo", "abril"))  
res: String = "marzo"
```

2. Escribir una función llamada *penultimo* que devuelva el penúltimo elemento de una lista.

Como ejemplo:

```
scala > penultimo(List ("enero", "febrero", "marzo", "abril"))  
res: String = "marzo"
```

3. Escribir una función *eliminaElemento* que elimine todos los elementos *e* que se le pase como argumento a la lista de enteros *ls*

Como ejemplo:

```
scala > eliminaElemento(2, List (1, 1, 2, 3, 2))  
res: List[Int] = List(1,1,3)
```

4. Escribir una función *cuadrado* que devuelva la lista de enteros pasados como argumento elevados al cuadrado

Como ejemplo:

```
scala > cuadrado(List(1, 1, 2, 3, 2))  
res: List[Int] = List(1, 1, 4, 9, 4)
```

5. Escribir una función *esPalindromo* que devuelva si una secuencia de caracteres es un palíndromo.

Se considera palíndromo sí:

- Cadena de caracteres vacía ("")
- Cadena con un caracter ("a")
- *awa* es palíndromo si *w* es palíndromo ("ala")
- *aw* y *wa* es palíndromo si *w* es palíndromo y *a* es un símbolo especial ("*!ojo*")
- Nada más es un palíndromo

Como ejemplo:

```
scala > esPalindromo("ala")  
res: Boolean = true
```

6. Escribir una función *trim*, que acepta una cadena de caracteres y elimina los caracteres que no sean letras en los extremos

Como ejemplo:

```
scala > trim("ala!")  
res: String = "ala"
```

7. Escribir la función *simplifica* que acepta una lista de números enteros, posiblemente repetidos y devuelve una lista en la que se han eliminado los elementos repetidos consecutivos y se han sustituido por unos sólo.

Como ejemplo:

```
scala > simplifica(List(2 2 2 3 3 2 2 4 5 5))  
res: List[Int] = List(2 3 2 4 5)
```