

# Ejercicios de Programación Funcional en Swift 6

## Closures

1. Escribe una closure que acepte un número entero y devuelva su cuadrado.
2. Crea una closure que tome un string y devuelva su longitud.
3. Implementa una closure que reciba dos números y devuelva el mayor.
4. Crea una closure que tome un array de enteros y devuelva un array con los números multiplicados por 2.
5. Implementa una closure que combine dos strings en una sola frase.

## .map

6. Usa `.map` para convertir un array de strings en sus longitudes.
7. Usa `.map` para transformar un array de enteros en su representación en formato string.
8. Aplica `.map` a un array de objetos y devuelve un array con un atributo específico de esos objetos.
9. Usa `.map` para convertir un array de fechas en strings formateados.
10. Usa `.map` para transformar un array de números en su valor absoluto.

## .filter

11. Usa `.filter` para obtener todos los números pares de un array.
12. Usa `.filter` para obtener todas las palabras que tienen más de 3 letras de un array de strings.
13. Aplica `.filter` para encontrar todas las tareas que están pendientes en un array de objetos.
14. Usa `.filter` para obtener todos los elementos de un array que son mayores que un valor dado.
15. Usa `.filter` para extraer los strings que contienen una letra específica.

## **.reduce**

16. Usa `.reduce` para sumar todos los números en un array.
17. Aplica `.reduce` para concatenar todos los strings de un array.
18. Usa `.reduce` para encontrar el valor máximo en un array de enteros.
19. Usa `.reduce` para contar la cantidad de elementos en un array que cumplen una condición.
20. Aplica `.reduce` para calcular el promedio de un array de números.

## **.allSatisfy**

21. Usa `.allSatisfy` para verificar si todos los elementos de un array son positivos.
22. Aplica `.allSatisfy` para comprobar si todas las palabras en un array tienen menos de 10 letras.
23. Usa `.allSatisfy` para verificar si todos los productos en un array son de una categoría específica.
24. Aplica `.allSatisfy` para comprobar si todos los números en un array son múltiplos de un número dado.
25. Usa `.allSatisfy` para verificar si todos los elementos en un array cumplen con una condición particular.

## **.contains**

26. Usa `.contains` para verificar si un array de strings incluye un nombre específico.
27. Aplica `.contains` para comprobar si un array de enteros tiene un valor específico.
28. Usa `.contains` para verificar si hay un objeto con una propiedad específica en un array.
29. Aplica `.contains` para comprobar si un string contiene una subcadena específica.
30. Usa `.contains` para verificar si un array de fechas incluye una fecha específica.

## **.drop**

31. Usa `.drop` para eliminar los primeros 3 elementos de un array.

- 32. Aplica `.drop` para eliminar los últimos 2 elementos de un array.
- 33. Usa `.drop` para eliminar los elementos hasta que se encuentre un valor específico.
- 34. Aplica `.drop` para ignorar los elementos que no cumplen con una condición hasta encontrar uno que sí.
- 35. Usa `.drop` para eliminar elementos mientras una condición se cumpla.

## **.prefix**

- 36. Usa `.prefix` para obtener los primeros 4 elementos de un array.
- 37. Aplica `.prefix` para obtener todos los elementos hasta encontrar un número específico.
- 38. Usa `.prefix` para obtener los primeros n caracteres de un string.
- 39. Aplica `.prefix` para limitar la cantidad de objetos en un array basado en una condición.
- 40. Usa `.prefix` para obtener los primeros n elementos de una colección filtrada.

## **.forEach**

- 41. Usa `.forEach` para imprimir cada elemento de un array de strings.
- 42. Aplica `.forEach` para incrementar el valor de cada elemento en un array de enteros.
- 43. Usa `.forEach` para crear un nuevo array a partir de una acción realizada en cada elemento.
- 44. Aplica `.forEach` para modificar propiedades de objetos en un array.
- 45. Usa `.forEach` para ejecutar una operación en cada elemento de un array y contar elementos.

## **.sorted**

- 46. Usa `.sorted` para ordenar un array de enteros en orden ascendente.
- 47. Aplica `.sorted` para ordenar un array de strings en orden alfabético.
- 48. Usa `.sorted` para ordenar un array de objetos basado en un atributo numérico.
- 49. Aplica `.sorted` para ordenar un array de fechas en orden cronológico.

50. Usa `.sorted` para ordenar un array de strings en orden inverso.

## Ejercicios Combinados

1. Usa `.map` para obtener los cuadrados de un array de números y luego usa `.filter` para quedarte solo con los números pares de esos cuadrados.
2. Aplica `.filter` a un array de strings para obtener solo los que tienen más de 3 letras y luego usa `.map` para convertir esos strings a mayúsculas.
3. Usa `.reduce` para sumar todos los números de un array y luego verifica si el resultado es mayor que 100 usando `.allSatisfy`.
4. Toma un array de objetos, usa `.filter` para obtener solo los que cumplen una condición, y luego usa `.map` para extraer un atributo específico de esos objetos.
5. Aplica `.drop` a un array de números para eliminar los primeros tres elementos y luego usa `.map` para multiplicar los elementos restantes por 10.
6. Usa `.prefix` para obtener los primeros 5 elementos de un array, luego usa `.forEach` para imprimir cada uno de esos elementos en una línea.
7. Usa `.sorted` para ordenar un array de enteros y luego utiliza `.map` para convertir los números ordenados a sus equivalentes en strings.
8. Aplica `.contains` para verificar si un array tiene un string específico y luego usa `.allSatisfy` para comprobar si todos los elementos del array son menores que una longitud determinada.
9. Usa `.filter` para obtener un array de productos en stock, luego usa `.reduce` para contar cuántos quedan en total.
10. Toma un array de fechas, usa `.sorted` para ordenarlas cronológicamente y luego aplica `.prefix` para obtener las tres más recientes.