

Ejercicio 1

Qué resultado arroja la siguiente expresión?

`(- 10 (* 2 3))`

Ejercicio 2

Escriba una expresión que filtre una secuencia de números retornando sólo aquellos que sean mayores a 5.

Ejercicio 3

Escriba una función que dado un mapa y una key retorne true si la key vale nil o false si vale otra cosa. La función debe cumplir con los tests que se muestran.

```
(def mapa {:a nil :b 1 :c 2})
```

```
(mapa-key-nula mapa :a) ;true
```

```
(mapa-key-nula mapa :b) ;false
```

```
(mapa-key-nula mapa :c) ;false
```

Ejercicio 4

Escriba una función (fibonacci) que calcule el n-ésimo número de Fibonacci:

$$F_0 = 1, F_1 = 1, F_{n+2} = F_{n+1} + F_n$$

Ejercicio 5

Escriba una función (sumar) que devuelva la suma de una secuencia de números. La misma debe satisfacer los tests que se muestran.

```
(sumar [1 2 3]) => 6
(sumar (list 0 -2 5 5)) => 8
(sumar #{4 2 1}) => 7
(sumar '(0 0 -1)) => -1
(sumar '(1 10 3)) => 14
```

Ejercicio 6

Escriba una función (ultimo) que devuelva el último elemento de una secuencia. La misma debe satisfacer los tests que se muestran.

```
(ultimo [1 2 3 4 5]) => 5
(ultimo '(5 4 3)) => 3
(ultimo ["b" "c" "d"]) => "d"
```

Ejercicio 7

Escriba una función (enesimo) que devuelva el enésimo elemento de una secuencia. La misma debe satisfacer los tests que se muestran.

```
(enesimo 2 '(4 5 6 7))
=> 6
(enesimo 0 [:a :b :c])
=> :a
(enesimo 1 [1 2 3 4])
=> 2
(enesimo 2 '([1 2] [3 4] [5 6]))
=> [5 6]
```

Ejercicio 8

Escriba una función (`separador`) que divida un array de elementos a partir de la posición dada. La misma debe satisfacer los tests que se muestran.

```
(separador 3 [1 2 3 4 5 6]) => [(1 2 3) (4 5 6)]
```

```
(separador 1 [:a :b :c :d]) => [(:a) (:b :c :d)]
```

```
(separador 2 [[1 2] [3 4] [5 6]]) => [[(1 2) [3 4]] ([5 6])]
```