

Introducción a la Programación

Augusto González Omahen

Primer Semestre de 2022

Clase práctica 15: Recursión

Ejercicio 1: Recursión sobre enteros

Implementemos las funciones usando **recursión**:

- ▶ *factorial* que recibe un número natural y devuelve su factorial:

$$factorial(3) \rightarrow 3 \times 2 \times 1 = 6$$

- ▶ *suma_digitos* que recibe un número natural y devuelve la suma de sus dígitos:

$$suma_digitos(325) \rightarrow 3 + 2 + 5 = 10$$

- ▶ *suma_impares* que recibe un número **n** natural y devuelve la suma de los primeros **n** impares:

$$suma_impares(4) \rightarrow 1 + 3 + 5 + 7 = 16.$$

Ejercicio 2a: Recursión sobre listas

Sea la función **sumar_n** que toma una lista de enteros y un elemento **n** entero, entonces suma **n** a cada elemento de la lista original.

```
#Por ejemplo, deber\ian pasar los siguientes tests:
```

```
assertEqual(sumar_n([8,4,-1], 3) , [11,7,2])  
assertEqual(sumar_n([13,11,31], -1) , [12,10,30])  
assertEqual(sumar_n([], 7) , [])
```

Pensemos entonces una forma de escribir al algoritmo de manera recursiva!

Ejercicio 2b: Recursión sobre listas

Sea la función **codificacion_inversa** que toma una lista no vacía de strings, devuelve la concatenación de los mismos en orden inverso.

```
#Por ejemplo, deber\ 'ian pasar los siguientes tests:

assertEqual(codificacion_inversa(['mo','rit','go','al']) ,
            'algoritmo')
assertEqual(codificacion_inversa(['un','amo','sh']) ,
            'shamoun')
assertEqual(codificacion_inversa(['']) , '')
```

Pensemos entonces una forma de escribir al algoritmo de manera recursiva!

Resumiendo un poco

- ▶ Empezamos a pensar de manera recursiva!
- ▶ Practicamos la estructura de una función recursiva:
caso base + paso recursivo

Con lo visto, ya pueden arrancar a resolver la décima y última Guía de Ejercicios.