

Funciones recursivas

Ejercicio: Obtener los primeros n elementos de la serie Fibonacci mediante recursividad

La serie de Fibonacci es una secuencia infinita de números enteros en la que cada número es la suma de los dos anteriores. La secuencia comienza con los números 0 y 1, y a partir de ahí, cada número sucesivo es la suma de los dos anteriores.

Ejemplo de la serie de Fibonacci:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

En este ejercicio, se define una función recursiva para generar los primeros n elementos de la serie de Fibonacci.

- La función se define para generar los primeros n elementos de la serie de Fibonacci y utiliza una lista **resultado** para almacenar los elementos calculados
- **Caso Base:**
 - Cuando n es 0, la función retorna la lista **resultado** con los elementos de la serie calculados hasta ese momento:
- **Caso Recursivo:**
 - Si la lista **resultado** está vacía, se añade 0, que es el primer número en la serie de Fibonacci.
 - Si la lista contiene un solo elemento, se añade 1, que es el segundo número en la serie.
 - Para listas con más de un elemento, se añade el siguiente número en la serie sumando los dos últimos números de la lista:

```
def fibonacci_n_elementos(n, resultado=[]):  
    # Caso base  
    if n == 0:  
        return resultado  
  
    # Caso recursivo  
    if len(resultado) == 0:  
        resultado.append(0)  
    elif len(resultado) == 1:  
        resultado.append(1)  
    else:  
        resultado.append(resultado[-1] + resultado[-2])  
    return fibonacci_n_elementos(n - 1, resultado)  
  
print(fibonacci_n_elementos(15))
```

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]