



# Clase 3: Funciones Recursivas en Python



## Introducción a la Recursividad

La **recursividad** es una técnica de programación en la que una función se llama a sí misma para resolver un problema. Cuando comencé a aprender los fundamentos de la programación, la recursividad fue uno de los conceptos más desafiantes. Sin embargo, al empezar a usar Python, pude apreciar su potencial.



## Ejemplo: Calcular el Factorial



### ¿Qué es el Factorial?

El **factorial** de un número entero  $(n)$  se define como el producto de todos los enteros desde  $(n)$  hasta 1. Se denota como  $(n!)$ .

**Ejemplo:**

- $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$



### Función Recursiva para el Factorial

Para calcular el factorial de  $(n)$ , usamos la siguiente fórmula:

$$n! = n \times (n-1)!$$

### Caso Base:

- El caso base es  $(0! = 1)$ .

### Implementación en Python

```
def factorial(n):  
    if n == 0: # Caso base  
        return 1  
    else:  
        return n * factorial(n - 1) # Llamada recursiva  
  
# Ejemplo de uso  
print(factorial(5)) # Salida: 120
```

### Ejemplo: Serie de Fibonacci

La **serie de Fibonacci** comienza con 0 y 1, y cada número siguiente es la suma de los dos anteriores.

#### Ejemplo:

- 0, 1, 1, 2, 3, 5, 8, 13, ...

### Función Recursiva para Fibonacci

#### Caso Base:

- $\text{Fibonacci}(0) = 0$
- $\text{Fibonacci}(1) = 1$

#### Recursión:

- $\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2)$

## Implementación en Python

```
def fibonacci(n):
    if n == 0:
        return 0 # Caso base
    elif n == 1:
        return 1 # Caso base
    else:
        return fibonacci(n - 1) + fibonacci(n - 2) # Llamada
recursiva

# Ejemplo de uso
print(fibonacci(5)) # Salida: 5
```

## Ejercicio: Sumar Números Naturales

 **Reto:** Crear una función para calcular la sumatoria de números naturales.

### Definición:

Si queremos sumar los primeros  $(n)$  números naturales, podemos usar la siguiente fórmula:

$$S(n) = n + (n-1) + (n-2) + \dots + 1$$

## Implementación en Python

```
def sumatoria(n):
    if n == 0: # Caso base
        return 0
    else:
        return n + sumatoria(n - 1) # Llamada recursiva
```

```
# Ejemplo de uso  
print(sumatoria(4)) # Salida: 10 (4 + 3 + 2 + 1)
```

## Consejos para la Práctica

- **Ejercicios de práctica:** Intenta resolver problemas de la vida real utilizando recursividad, como la búsqueda en estructuras de datos (árboles, listas, etc.).
- **Revisa:** Asegúrate de entender cada parte de las funciones recursivas que implementes.