

# ESTRUTURA DE DADOS

MÓDULO 07 | AULA 09





# RECURSÃO

## Introdução

Recursão é um conceito fundamental na programação, onde uma função chama a si mesma para resolver um problema. Este método pode simplificar significativamente a solução de problemas complexos, dividindo-os em instâncias menores e mais gerenciáveis.

## Conceitos Básicos

- Recursão: Uma técnica de programação onde uma função chama a si mesma.
- Caso Base: A condição sob a qual a função retorna um valor sem fazer novas chamadas a si mesma, evitando assim um loop infinito.
- Caso Recursivo: A parte da função que inclui a chamada a si mesma, destinada a aproximar o problema da solução.

## Pilha de Chamadas

A execução de funções recursivas utiliza uma pilha de chamadas, onde cada chamada da função é colocada na pilha até que o caso base seja alcançado. Após alcançar o caso base, as funções começam a retornar seus valores e a pilha de chamadas é desfeita.



## Exemplo #01

O fatorial de um número  $n$ , denotado por  $n!$ , é o produto de todos os inteiros positivos menores ou iguais a  $n$ . A função recursiva abaixo calcula o fatorial:

```
def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n - 1)
```

## Exemplo #02

Na sequência de Fibonacci, cada número é a soma dos dois precedentes, com os dois primeiros números sendo 0 e 1. A implementação recursiva é:

```
def fibonacci(n):  
    if n <= 1:  
        return n  
    else:  
        return fibonacci(n - 1) + fibonacci(n - 2)
```



# RECURSÃO

## Recursão em Estruturas de Dados

Recursão é particularmente útil em estruturas de dados como árvores e grafos, permitindo operações como travessias e buscas de maneira intuitiva e eficiente.

- **Travessia em Árvores Binárias:** A recursão facilita a implementação de travessias inordem, preordem e pós-ordem em árvores binárias, permitindo processar os nós de uma árvore de maneira sistemática.
- **Busca em Grafos:** Algoritmos como a busca em profundidade (DFS) utilizam recursão para explorar todos os vértices de um grafo, seguindo cada caminho até o fim antes de retroceder.

## Problemas Comuns e Soluções com Recursão

Embora a recursão ofereça soluções elegantes, ela pode levar a problemas como estouro de pilha e cálculos redundantes. Técnicas como memoização, que armazena resultados de chamadas de função, podem otimizar o desempenho de funções recursivas.