

**INSTITUTO
FEDERAL**

Norte de Minas Gerais

TÉCNICA DE PROGRAMAÇÃO DIVISÃO E CONQUISTA

APRESENTADO POR: CARLOS DANIEL

DIVISÃO E CONQUISTA

- **Definição: Técnica que resolve um problema ao dividi-lo em subproblemas menores e mais simples.**

- **Três passos fundamentais:**

1. Dividir

2. Conquistar (resolver recursivamente)

3. Combinar

RECURSIVIDADE

- **Definição: Função que chama a si mesma para resolver uma parte do problema.**
- **Elementos essenciais:**
 - **Caso base: condição de parada**
 - **Chamada recursiva: resolve uma versão menor do problema**
 - **Pilhas de funções**

ITERAÇÃO VS RECURSÃO

- **Iteração:**
- **Usa laços (for, while)**
- **Geralmente mais eficiente em termos de memória**
- **Recursão:**
- **Mais intuitiva em certos casos**
- **Usa a pilha de chamadas**

CALCULAR FATORIAL

fatorial

$$n! = 1 \times \dots \times n$$

CALCULAR FATORIAL

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

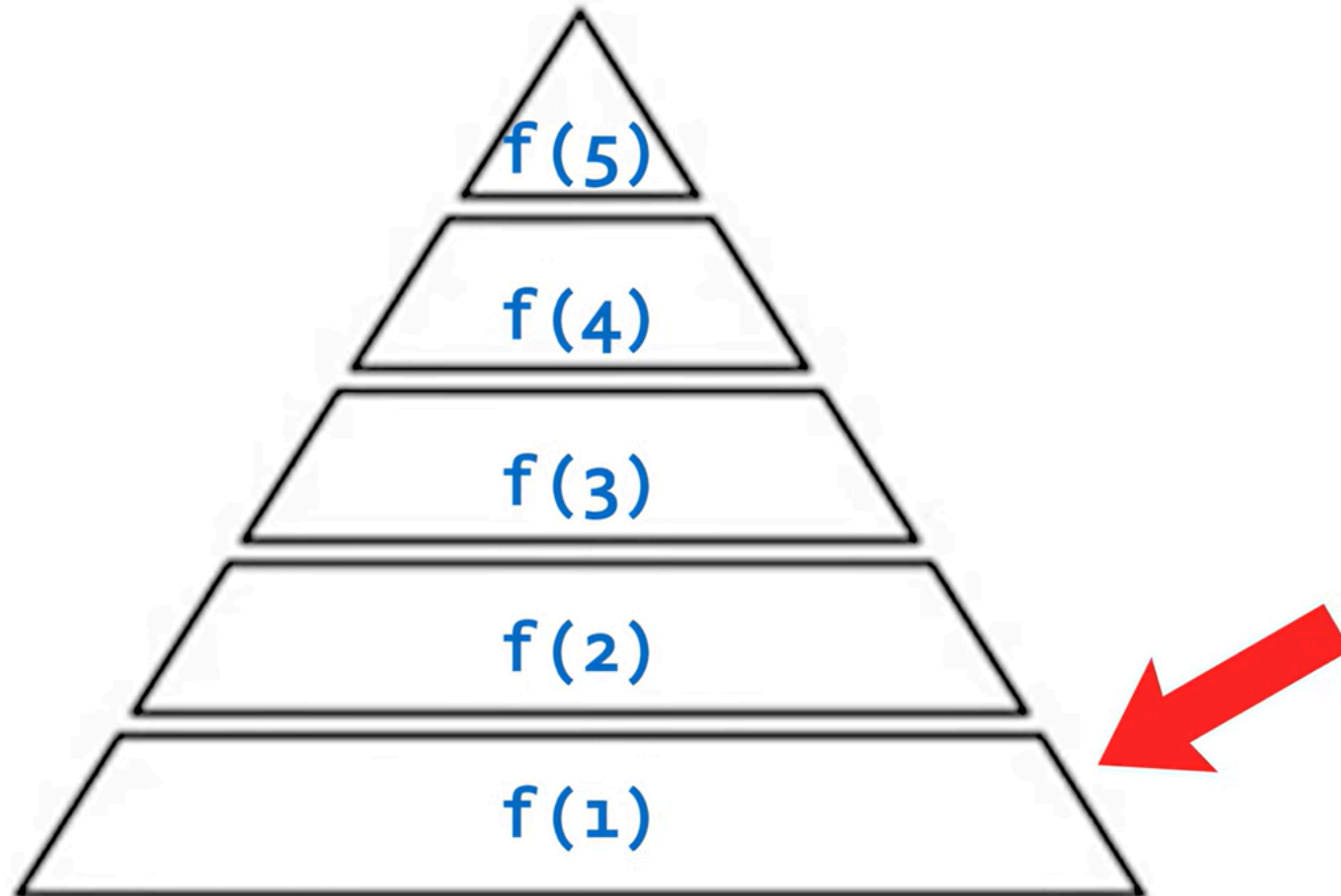
CALCULAR FATORIAL DE FORMA DE ITERAÇÃO

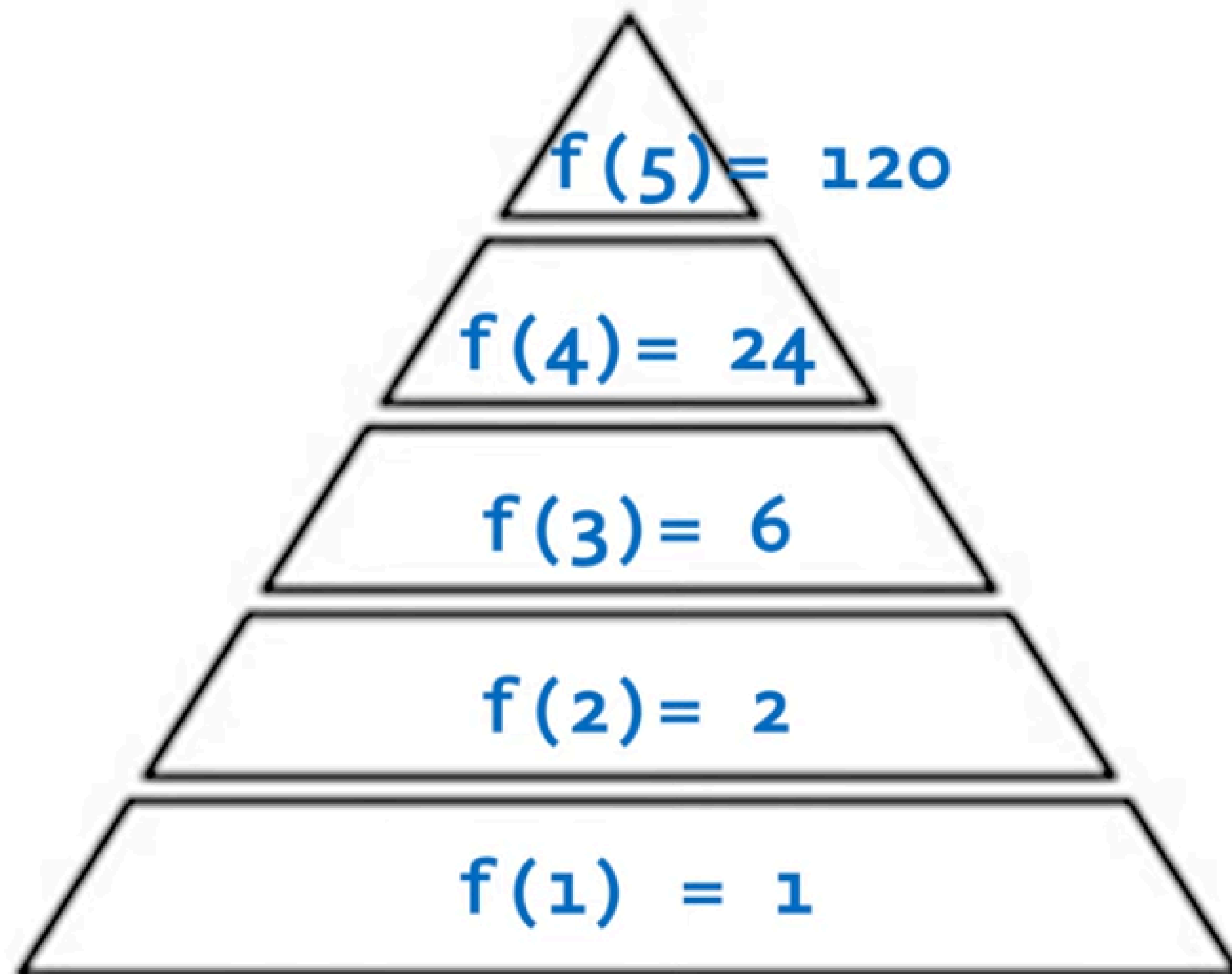
```
int fatorial_iterativo(int n) {  
    int resultado = 1;  
    for (int i = 1; i <= n; i++) {  
        resultado *= i;  
    }  
    return resultado;  
}
```

CALCULAR FATORIAL DE FORMA DE RECURSÃO

```
int fatorial_recursivo(int n) {  
    if (n == 0 || n == 1)  
        return 1; // Casos base  
    else  
        return n * fatorial_recursivo(n - 1); // Chamada recursiva  
}
```


CASO BASE







VANTAGENS E DESVANTAGENS

- **Desempenho:** Funções recursivas geralmente são mais lentas por criarem várias chamadas de si mesmas.
- **Uso de memória:** A recursão consome mais memória devido à pilha de chamadas.
- **Clareza:** Recursão pode tornar o código mais simples e fácil de entender em certos problemas.
- **Eficiência:** Iterações costumam ser mais rápidas e eficientes.
- **Complexidade:** Código iterativo tende a ser mais complexo, mas com melhor desempenho.

OBRIGADO!