

# Recursividade



**João Frederico Roldan Viana**

[jfredrv@gmail.com](mailto:jfredrv@gmail.com)

(85)99231.2777



# Agenda

---

## ■ Recursividade

- Definições
- Exemplo
- Cuidado
- Desvantagens
- Vantagens

# Recursividade

## ■ Definições

- Referência circular.
- Chamar uma função dentro dela mesmo.
- O objeto recursivo é definido em termos de si próprio.
- Exemplo: Fatorial ( $n!$ )
  - $\text{fat}(n) = 1$ , quando  $n = 0$
  - $\text{fat}(n) = n * \text{fat}(n - 1)$ , quando  $n > 0$

# Recursividade

## ■ Exemplo

### ➤ Fatorial Não recursivo

```
fatnr(int n)
{
    int i, fat = 1;
    for(i = 2; i <= n; i++)
        fat = fat * i;
    return fat;
}
```

### ➤ Fatorial Recursivo

```
fatr(int n)
{
    if (n == 0)
        return 1;
    else
        return fatr(n-1)*n;
}
```

# Recursividade

---

## ■ Cuidado

- Deve existir uma condição, antes da chamada recursiva, que retorne sem realizá-la. Senão a função nunca retorna um valor, ou seja, ficará em um laço (loop) infinito.

# Recursividade

---

## ■ Desvantagens

- A maioria não minimiza o tamanho do código, nem melhora a utilização da memória
- Pode ser executada mais lentamente, devido as suas repetidas chamadas
- Muitas chamadas podem ocasionar estouro de pilha

# Recursividade

---

## ■ Vantagens

- Cria versões mais claras e simples de alguns algoritmos
- Alguns problemas (principalmente em IA) resultam em soluções recursivas
- Algumas pessoas pensam melhor recursivamente