Recursão: conceitos básicos

- Consiste em reduzirmos um determinado problema a uma instância menor do mesmo problema

- Se o problema for suficientemente pequeno, resolva-o diretamente
- Senão reduza-o a uma instância menor do mesmo problema e execute a verificação acima novamente

Recursão: conceitos básicos

- Exemplo: arrecadação de 1500 reais em doações

Recursão: conceitos básicos

- Exemplo clássico: cálculo do fatorial

0! = 1

1! = 1

2! = 2 * 1 = 2

3! = 3 * 2 * 1 = 6

…

n! = n * … * 1

$$fatorial(n) = \begin{cases} 1 & \text{se } n = 0 \\ n \times fatorial(n-1) & \text{se } n > 0 \end{cases}$$

Recursão: conceitos básicos

fatorial (3) = 3 * fatorial (3 - 1)

= 3 * fatorial (2)
= 3 * 2 * fatorial (2 - 1)
= 3 * 2 * fatorial (1)
= 3 * 2 * 1 * fatorial (1 - 1)
= 3 * 2 * 1 * fatorial (0)
= 3 * 2 * 1 * 1
= 6

$$fatorial(n) = \begin{cases} 1 & \text{se } n = 0 \\ n \times fatorial(n-1) & \text{se } n > 0 \end{cases}$$

Recursão: conceitos básicos

```
/* função recursiva para cálculo do fatorial */

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

Recursão: conceitos básicos

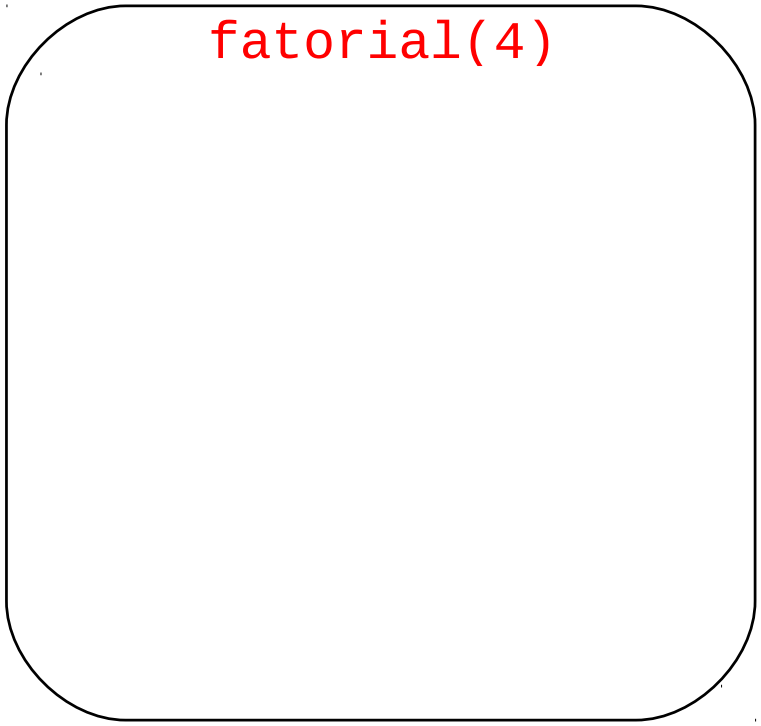
- Para cada chamada de uma função, recursiva ou não, os parâmetros e as variáveis são empilhados na pilha de execução.

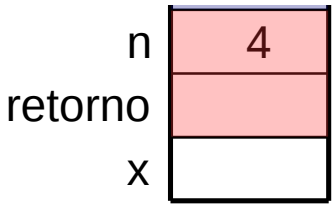
- Então, cada chamada recursiva cria um ambiente local na memória, fazendo com que as variáveis sejam independentes entre si.

Recursão

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

x = fat(4);
```

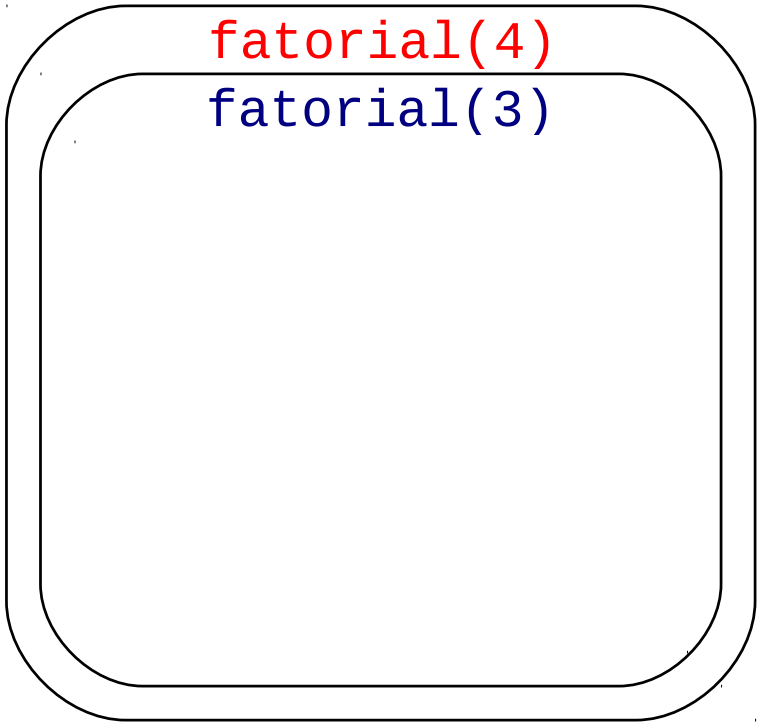


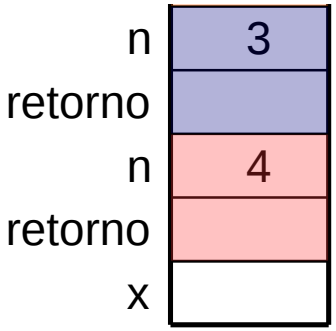


Recursão

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

x = fat(4);
```

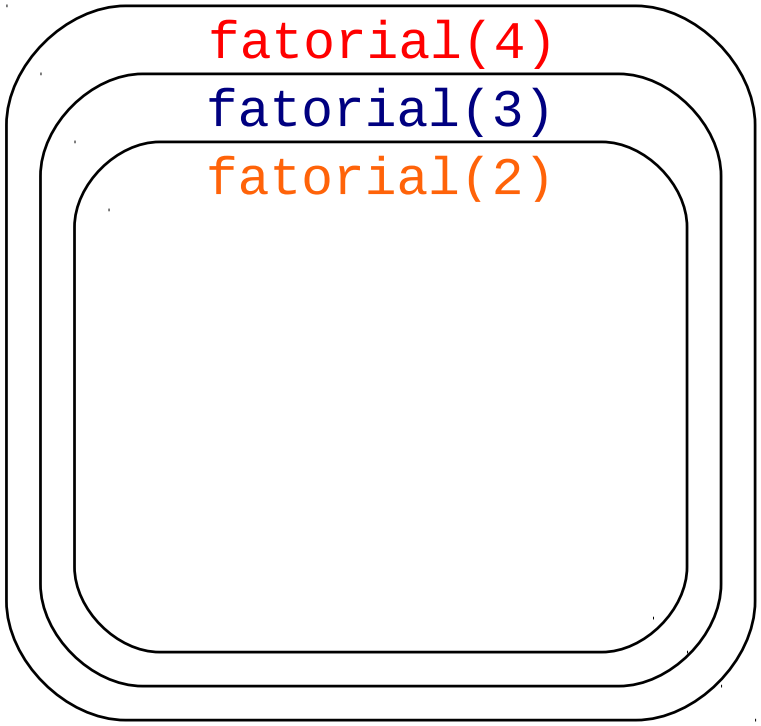


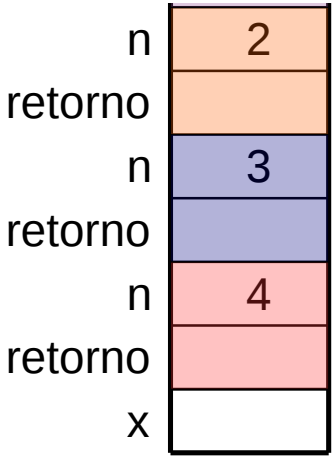


Recursão

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

x = fat(4);
```

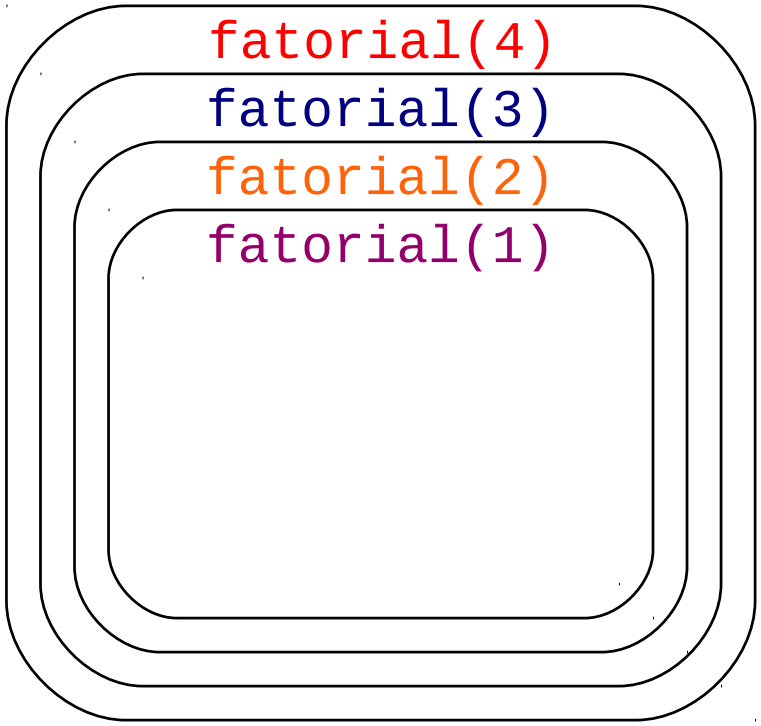


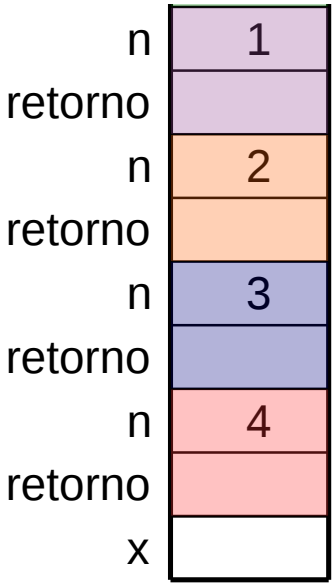


Recursão

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

x = fat(4);
```

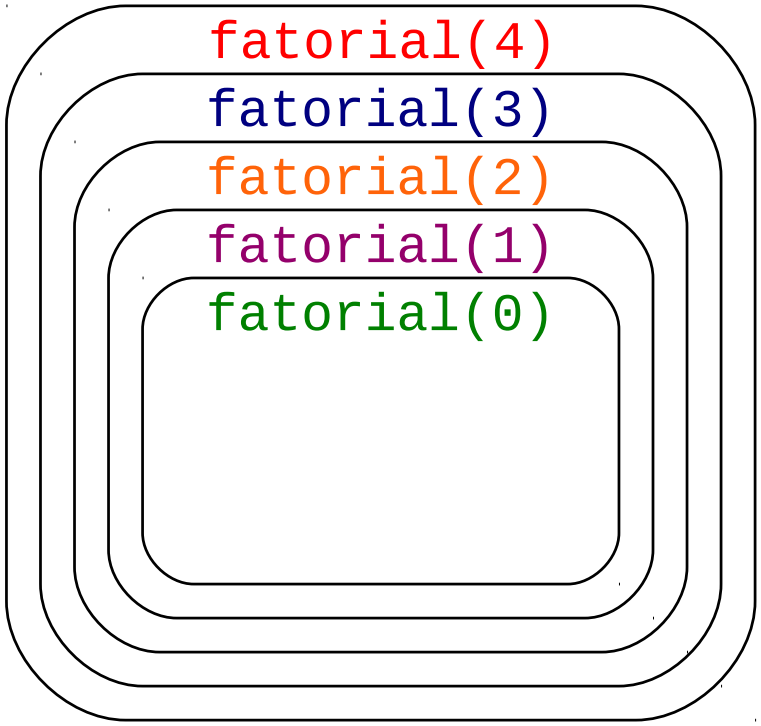


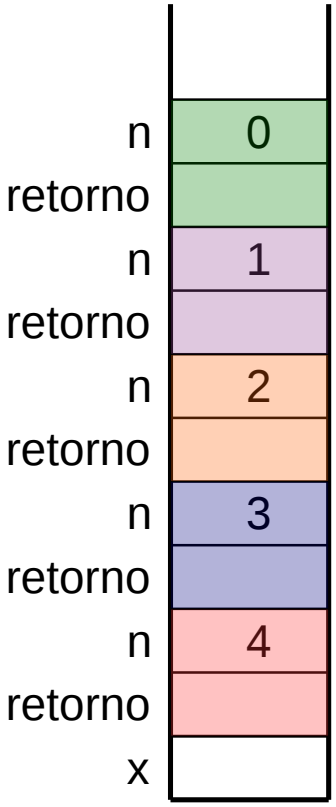


Recursão

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

x = fat(4);
```

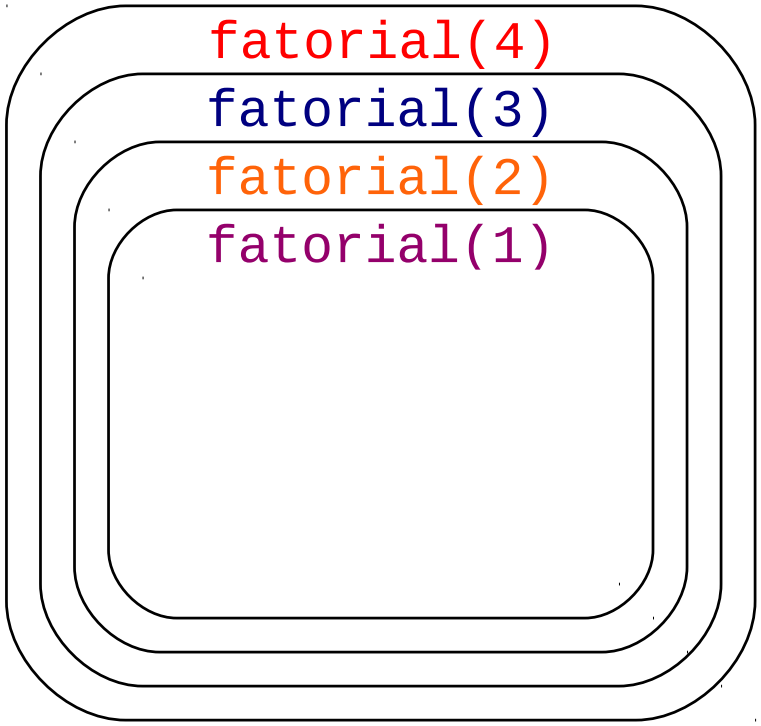


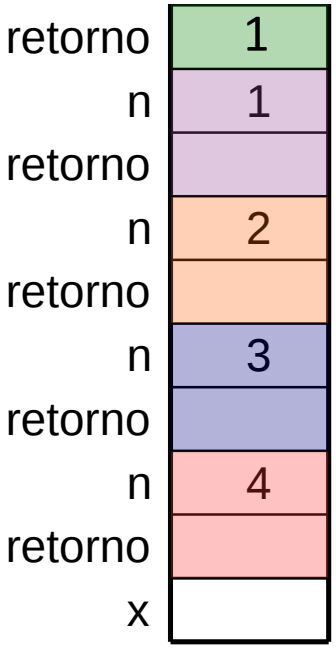


Recursão

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

x = fat(4);
```

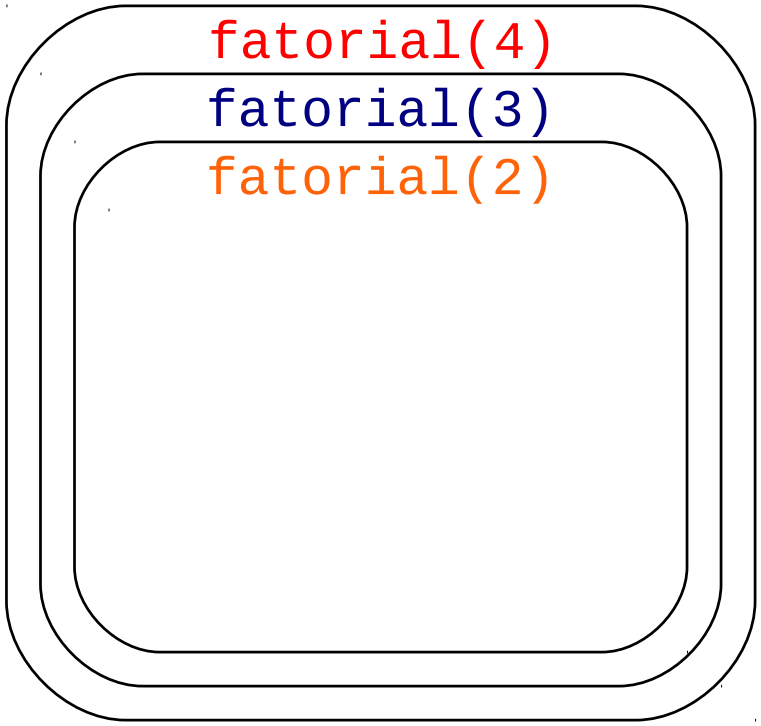


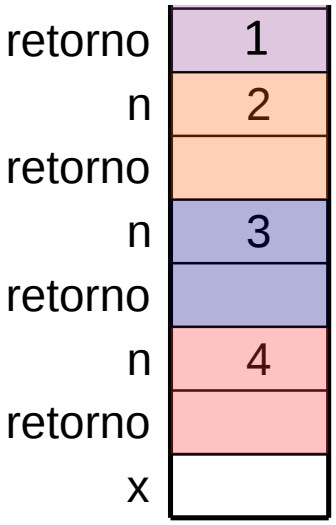


Recursão

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

x = fat(4);
```

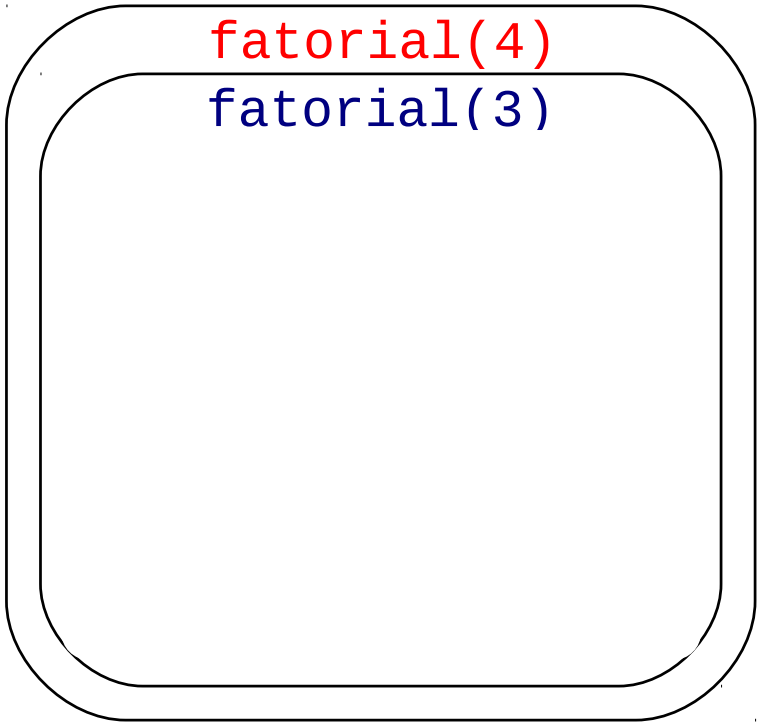


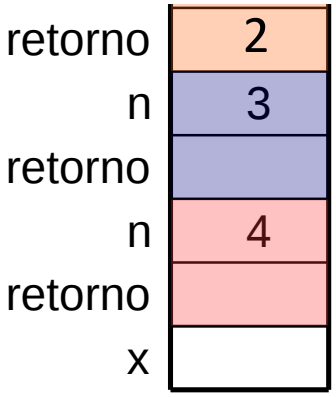


Recursão

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

x = fat(4);
```

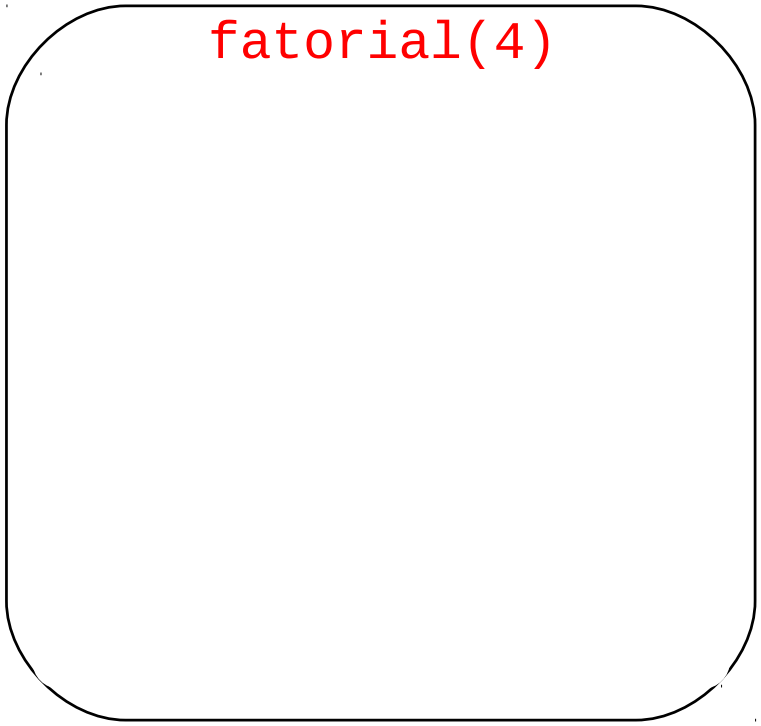


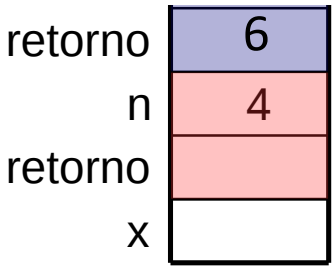


Recursão

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

x = fat(4);
```

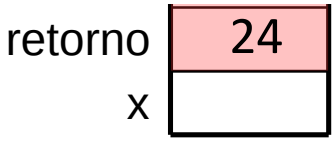




Recursão

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

x = fat(4);
```



Recursão

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

x = fat(4);
```

x | 24