



WWW.UNICARIOCA.EDU.BR

# Algoritmos II

## Unidade 2 - Recursividade



Prof<sup>a</sup> Giselle Batalha

**MELHOR CENTRO UNIVERSITÁRIO DO RIO DE JANEIRO!**

## Recursividade

É o mecanismo de programação no qual uma definição de função ou de outro objeto refere-se ao próprio objeto sendo definido.

Assim, função recursiva é uma função que é definida em termos de si mesma.

Recursividade é o mecanismo básico para repetições nas linguagens funcionais.

São sinônimos: recursividade, recursão, recorrência.





Estratégia para a definição recursiva de uma função:

- 1 - dividir o problema em problemas menores do mesmo tipo
- 2 - resolver os problemas menores (dividindo-os em problemas ainda menores, se necessário)
- 3 - combinar as soluções dos problemas menores para formar a solução final

Ao dividir o problema sucessivamente em problemas menores eventualmente os casos simples são alcançados:

- I- não podem ser mais divididos
- II - suas soluções são definidas explicitamente



De modo geral, uma definição de função recursiva é dividida em duas partes:

- ❑ Há um ou mais **casos base** que dizem o que fazer em situações simples, onde não é necessária nenhuma recursão. Nestes casos a resposta pode ser dada de imediato, sem chamar recursivamente a função sendo definida. Isso garante que a recursão eventualmente possa parar.
- ❑ Há um ou mais **casos recursivos** que são mais gerais, e definem a função em termos de uma chamada mais simples a si mesma

# FATORIAL SEM RECURSIVIDADE

Algoritmo "fatorial"

var

n,contador,fatorial: inteiro

inicio

escreval(" Cálculo do fatorial de um numero")

escreva("Entre com um inteiro nao-negativo: ")

leia(n)

contador <- n

fatorial<-1

enquanto contador >= 1 faca

fatorial <- fatorial\*contador

contador <- contador-1

fimenquanto

escreval("N = ",n," Fatorial = ",fatorial)

Fimalgoritmo

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
int n, x, Contador, fatorial=1;
```

```
printf(" Cálculo do fatorial de um numero");
```

```
printf ("Entre com um inteiro nao-negativo: ");
```

```
scanf("%d",&n);
```

```
Contador = n;
```

```
do
```

```
{
```

```
fatorial = fatorial*Contador;
```

```
Contador = Contador-1;
```

```
}
```

```
while(Contador >= 1);
```

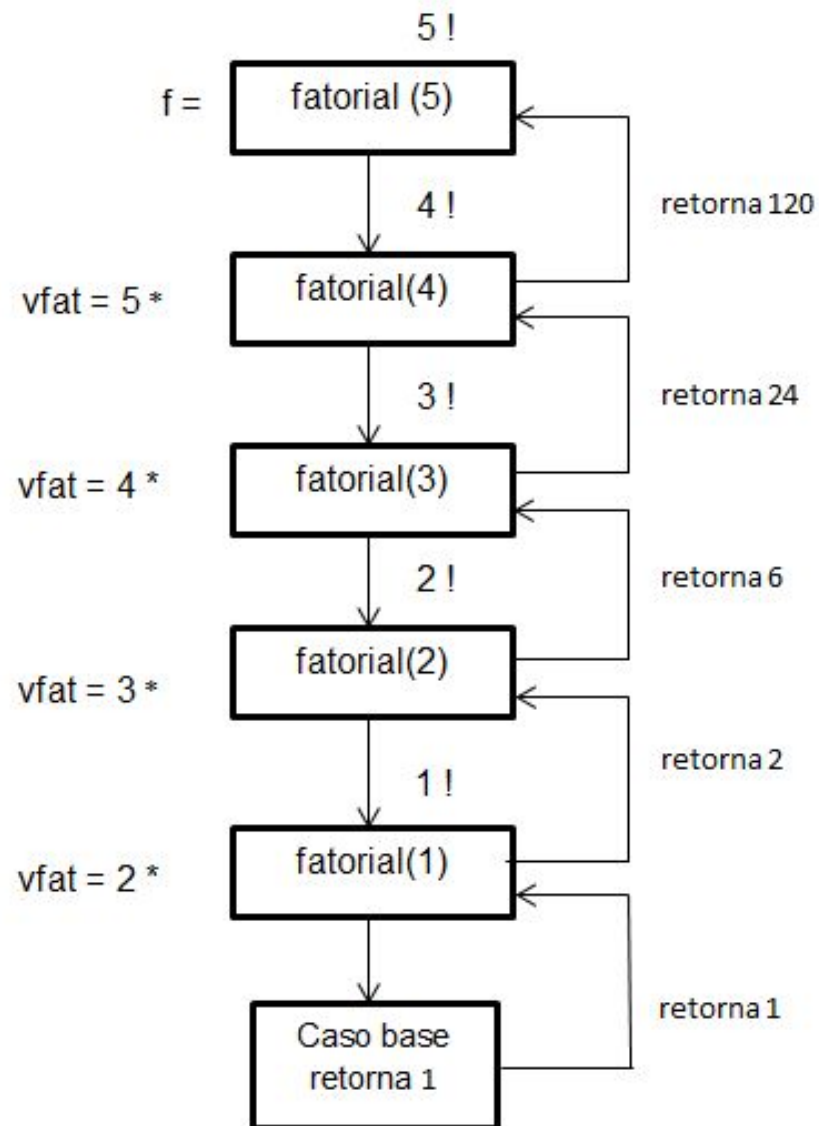
```
printf ("%d! = %d ",n,fatorial);
```

```
}
```

## Aplicando a função fatorial:

```
fatorial 6
↪ fatorial 5 * 6
↪ (fatorial 4 * 5) * 6
↪ ((fatorial 3 * 4) * 5) * 6
↪ (((fatorial 2 * 3) * 4) * 5) * 6
↪ ((((fatorial 1 * 2) * 3) * 4) * 5) * 6
↪ ((((((fatorial 0 * 1) * 2) * 3) * 4) * 5) * 6)
↪ ((((((1 * 1) * 2) * 3) * 4) * 5) * 6)
↪ (((((1 * 2) * 3) * 4) * 5) * 6)
↪ (((2 * 3) * 4) * 5) * 6
↪ ((6 * 4) * 5) * 6
↪ (24 * 5) * 6
↪ 120 * 6
↪ 720
```





Algoritmo "fatorial"

**funcao fatorial (N:inteiro): inteiro**

**inicio**

**se (N=0) entao**

**retorne 1 //caso base**

**senao**

**retorne N\*fatorial(N-1) //caso recursivo**

**fimse**

**fimfuncao**

var

n:inteiro

inicio

escreval(" Cálculo do fatorial de um numero")

escreva ("Entre com um inteiro nao-negativo: ")

leia(n)

escreval ("N = ",n," Fatorial =",fatorial(n))

fimalgoritmo

```
#include<stdio.h>
```

```
int fatorial (int N)
```

```
{
```

```
    if (N==0)
```

```
        return 1; //caso base
```

```
    else
```

```
        return N*fatorial(N-1); //caso recursivo
```

```
}
```

```
main()
```

```
{
```

```
    int n;
```

```
    printf(" Cálculo do fatorial de um numero");
```

```
    printf ("Entre com um inteiro nao-negativo: ");
```

```
    scanf("%d",&n);
```

```
    printf ("%d! = %d ",n,fatorial(n));
```

```
}
```



[WWW.UNICARIOCA.EDU.BR](http://WWW.UNICARIOCA.EDU.BR)



**MELHOR CENTRO UNIVERSITÁRIO DO RIO DE JANEIRO!**

Fonte: MEC