



# MÉTODOS

FAPESC – DESENVOLVEDORES PARA TECNOLOGIA DA INFORMAÇÃO

HABNER FABRÍCIO BOESING  
[habner.boesing@unoesc.edu.br](mailto:habner.boesing@unoesc.edu.br)



# OBJETIVOS

- Métodos
  - Procedimentos
  - Funções

# MÉTODOS

- Métodos representam um conjunto de instruções (subprograma), geralmente representando rotinas que podem se repetir em um programa.
- Diferem-se dos laços de repetição, pois neste caso, os comando podem ser invocados a qualquer momento, e não precisam de condições de parada, nem da obrigação da indicação do número de vezes que se deseja o comando.
- O principal objetivo é evitar a reescrita de comandos, caso exista algum comando padrão que possa ser acionado diversas vezes durante a execução de um algoritmo.

## Exemplo: CALCULADORA

- Os processos abaixo podem se repetir diversas vezes. Uma soma entre dois números pode ser repetida diversas vezes, mudando apenas os números utilizados em cada cálculo.

- Soma
- Subtração
- Multiplicação
- Divisão



# MÉTODOS – PROCEDIMENTOS E FUNÇÕES

- Os métodos são inseridos após a declaração das variáveis e antes do início do algoritmo principal e podem ser divididos em duas subcategorias:
  - **Procedimento:** quando o subprograma não retorna nenhum valor;
  - **Função:** quando o subprograma retorna um valor e por isso deve se especificar seu tipo.

## Procedimento:

```
procedimento soma
var aux: inteiro
inicio
// n, m e res são variáveis globais
aux <- n + m
res <- aux
fimprocedimento
```

## Função:

```
funcao soma: inteiro
var aux: inteiro
inicio
// n, m e res são variáveis globais
aux <- n + m
retorne aux
fimfuncao
```

No programa principal deve haver os seguintes comandos:

```
n <- 4
m <- -9
soma
escreva(res)
```

No programa principal deve haver os seguintes comandos:

```
n <- 4
m <- -9
res <- soma
escreva(res)
```

# PROCEDIMENTO SEM PARÂMETRO

- Exemplo:
- Operações básicas de uma calculadora.
- Por ser um procedimento o cálculo é realizado dentro do subprograma, mas não é retornado nenhum resultado de forma automática, sendo necessário solicitar a exibição da resposta fora do programa.

```
1 Algoritmo "PROCEDIMENTO SEM PARÂMETRO"
2
3 Var
4   num1,num2, resposta: real
5
6 procedimento soma
7 inicio
8   resposta <- num1 + num2
9 fimprocedimento
10
11 procedimento sub
12 inicio
13   resposta <- num1 - num2
14 fimprocedimento
15
16 procedimento mult
17 inicio
18   resposta <- num1 * num2
19 fimprocedimento
20
21 procedimento divisao
22 inicio
23   resposta <- num1 + num2
24 fimprocedimento
25
26 Inicio
27   num1 <- 10
28   num2 <- 5
29   soma
30   escreval(resposta)
31   sub
32   escreval(resposta)
33   mult
34   escreval(resposta)
35   divisao
36   escreval(resposta)
37 Fimalgoritmo
```

# PROCEDIMENTO COM PARÂMETRO

- Exemplo:
- Operações básicas de uma calculadora.
- Neste caso com a passagem de parâmetros, dentro dos procedimentos utilizamos variáveis locais ao invés de globais, as quais clonam o valor da variável global, servindo como substitutas enquanto os cálculos são realizados dentro do procedimento.
- O fato de variáveis globais serem acessadas por qualquer função é algo perigoso de se administrar, principalmente em códigos grandes. Lembrando que variáveis globais mantêm o seu valor durante toda a execução da aplicação.
- Imagine uma variável global que é utilizada por 15 procedimentos diferentes. Ficaria extremamente complicado saber seu valor atual no momento de efetuar uma verificação de código. Por isso utilizamos variáveis locais no lugar delas.

```
1 Algoritmo "PROCEDIMENTO COM PARÂMETRO"
2
3 Var
4   num1,num2, resposta: real
5
6 procedimento soma (x,y: real)
7 inicio
8   resposta <- x + y
9 fimprocedimento
10
11 procedimento sub (x,y: real)
12 inicio
13   resposta <- x - y
14 fimprocedimento
15
16 procedimento mult (x,y: real)
17 inicio
18   resposta <- x * y
19 fimprocedimento
20
21 procedimento divisao (x,y: real)
22 inicio
23   resposta <- x / y
24 fimprocedimento
25
26 Inicio
27   num1 <- 10
28   num2 <- 5
29   soma(num1,num2)
30   escreval(resposta)
31   sub(num1,num2)
32   escreval(resposta)
33   mult(num1,num2)
34   escreval(resposta)
35   divisao(num1,num2)
36   escreval(resposta)
37 Fimalgoritmo
```

# FUNÇÃO SEM PARÂMETRO

- Exemplo:
- Operações básicas de uma calculadora.
  - Aqui a diferença é que agora o valor do cálculo é retornado ao final do métodos, caracterizando-o assim como uma função.
  - Além disso com uma função, o valor é atribuído à resposta fora do método (e não mais dentro, como era com o procedimento), recebendo o valor de retorno calculado pela variável auxiliar dentro da função.

```
1 Algoritmo "FUNÇÃO SEM PARÂMETRO"
2 Var
3   num1,num2, resposta: real
4
5 funcao soma: real
6   var aux: real
7   inicio
8     aux <- num1 + num2
9     retorne aux
10 fimfuncao
11
12 funcao sub: real
13   var aux: real
14   inicio
15     aux <- num1 - num2
16     retorne aux
17 fimfuncao
18
19 funcao mult: real
20   var aux: real
21   inicio
22     aux <- num1 * num2
23     retorne aux
24 fimfuncao
25
26 funcao divisao: real
27   var aux: real
28   inicio
29     aux <- num1 + num2
30     retorne aux
31 fimfuncao
32
33 Inicio
34   num1 <- 10
35   num2 <- 5
36   resposta <- soma
37   escreval(resposta)
38   resposta <- sub
39   escreval(resposta)
40   resposta <- mult
41   escreval(resposta)
42   resposta <- divisao
43   escreval(resposta)
44 Fimalgoritmo
```

# FUNÇÃO COM PARÂMETRO

- Exemplo:
- Operações básicas de uma calculadora.
  - Aqui a diferença é que agora o valor do cálculo é retornado ao final do métodos, caracterizando-o assim como uma função.
  - Além disso com uma função, o valor é atribuído à resposta fora do método (e não mais dentro, como era com o procedimento), recebendo o valor de retorno calculado pela variável auxiliar dentro da função.

```
1 Algoritmo "FUNÇÃO COM PARÂMETRO"
2 Var
3   num1,num2, resposta: real
4
5 funcao soma(x,y:real): real
6   var aux: real
7   inicio
8     aux <- x + y
9     retorne aux
10 fimfuncao
11
12 funcao sub(x,y:real): real
13   var aux: real
14   inicio
15     aux <- x - y
16     retorne aux
17 fimfuncao
18
19 funcao mult(x,y:real): real
20   var aux: real
21   inicio
22     aux <- x * y
23     retorne aux
24 fimfuncao
25
26 funcao divisao(x,y:real): real
27   var aux: real
28   inicio
29     aux <- x + y
30     retorne aux
31 fimfuncao
32
33 Inicio
34   num1 <- 10
35   num2 <- 5
36   resposta <- soma(num1,num2)
37   escreval(resposta)
38   resposta <- sub(num1,num2)
39   escreval(resposta)
40   resposta <- mult(num1,num2)
41   escreval(resposta)
42   resposta <- divisao(num1,num2)
43   escreval(resposta)
44 Fimalgoritmo
```



# ATIVIDADES PRÁTICAS

- 1) Faça uma função que recebe um valor inteiro e verifica se o valor é positivo ou negativo. A função deve retornar um valor booleano.
- 2) Escreva uma função que recebe por parâmetro um valor inteiro e positivo N e retorna o valor de S.

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{N}.$$

- 3) Escreva uma função que recebe as 3 notas de um aluno por parâmetro e uma letra. Se a letra for A o procedimento calcula a média aritmética das notas do aluno, se for P, a sua média ponderada (pesos: 5, 3 e 2). Ao final a função deve retornar a média calculada.
- 4) Faça um procedimento que recebe, por parâmetro, um valor N e calcula e escreve a tabuada de 1 até N.