



# Procedimentos

**B**em-vindo ao estudo sobre Procedimentos! Essa aula ajudará na compreensão de alguns conceitos que são importantes no contexto de programação. Vamos conhecer melhor esses conceitos?

Modularização é a divisão de tarefas. Isso significa que o programa é dividido em partes ou módulos. Estes módulos são blocos de instruções que realizam tarefas específicas. Uma vez carregado, o módulo pode ser executado quantas vezes for necessário. Além disso, pode ser usado para economizar espaço e tempo de programação, já que pode ser chamado em várias partes de um mesmo programa (MANZANO; OLIVEIRA, 2012).

Cada módulo, além de ter acesso às variáveis do programa (variáveis globais), pode ter suas próprias variáveis (variáveis locais), que existem apenas durante sua chamada (RIBEIRO, 2019).

## Algumas vantagens na utilização de módulos

- Dividir e estruturar um algoritmo em partes logicamente coerentes;
- Facilidade de testar os trechos em separado;
- Evitar repetição do código-fonte;
- Maior legibilidade de um algoritmo.



## Tipos de subprogramas: Procedimentos e Funções

Neste módulo será mostrado o subprograma Procedimento.

### Procedimento

Procedimentos são estruturas que agrupam um conjunto de comandos, que são executados quando chamados no decorrer do algoritmo (MANZANO; OLIVEIRA, 2012).

Como um exemplo prático, em VisuAlg, procedimento pode ser definido como subprograma que não retorna nenhum valor. Sua declaração geralmente está no começo do algoritmo e sua sintaxe está descrita abaixo.

### Estrutura

**procedimento** <nome-de-procedimento> [( <sequência-de-declarações-de-parâmetros> )]

// Seção de Declarações de Variáveis Internas

**inicio**

// Seção de Comandos

**fimprocedimento**

### Exemplo

**PROCEDIMENTO** olaMundo

**VAR**

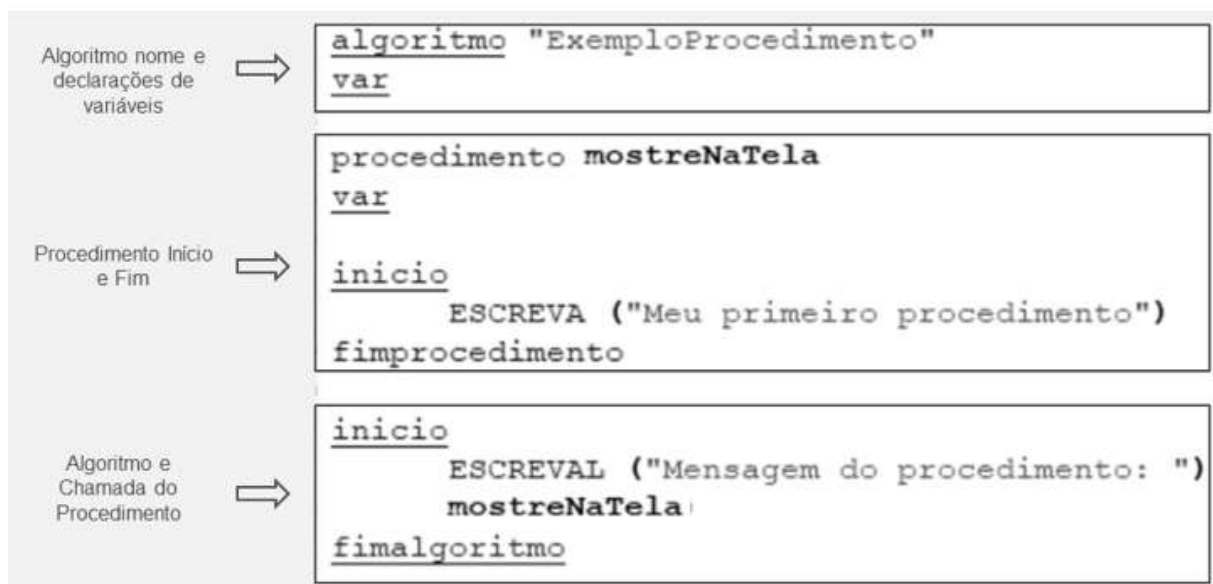
## INICIO



**ESCREVA** ("Olá mundo do procedimento!")

## FIMPROCEDIMENTO

### Exemplo Completo (Figura 1)




### Exemplo Prático

Para executar o pseudocódigo utilize o Visualg. No ambiente disponibilizado pela Faculdade Descomplica, basta acessar o ícone do Visualg (Figura 2).



Escreva o algoritmo em pseudocódigo, mostrado na Figura 3, na Área de Algoritmos da ferramenta.

```
Algoritmo "ProcedimentoExemp"   
procedimento soma  
var  
  aux: inteiro  
inicio  
  aux <- n + m  
  res <- aux  
fimprocedimento  
Var  
res, n, m: inteiro  
Inicio  
  n <- 4  
  m <- -9  
  soma  
  escreva(res)  
Fimalgoritmo
```

Para executar o algoritmo, clique no ícone "Executar", mostrado na Figura 4, ou F9 do seu teclado.



Microsoft, mostrando a conflituosa relação de Steve Jobs (Noah Wyle) e Bill Gates (Anthony Michael Hall) nos bastidores.



## Referência Bibliográfica

- GUEDES, S. (Org.). **Lógica de programação algorítmica**. Pearson: 2014.
- MANZANO, J. A. N. G.; OLIVEIRA, J. F. **Estudo Dirigido de Algoritmos**. 15. ed. São Paulo: Érica, 2012
- PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados, com aplicações em Java**. Pearson: 2016.
- RIBEIRO, J. A. **Introdução à programação e aos algoritmos**. 1. ed. Rio de Janeiro: LTC, 2019

## Atividade Prática – Aula 9

**Título da Prática:** Multiplicação como uso de Procedimento

**Aulas Envolvidas nesta Prática:** Procedimentos

**Objetivos:** Praticar lógica de programação e desenvolvimento de algoritmos.

**Materiais, Métodos e Ferramentas:** Para realizar este exercício, vamos utilizar Visualg para testar o algoritmo proposto no desenvolvimento da prática em questão.

## Atividade Prática



Com os conhecimentos adquiridos até agora, desenvolva um algoritmo em pseudocódigo que multiplique 2 números digitados pelo usuário. Mostre o resultado na tela. (Use procedimento para o cálculo)

Após desenvolver seu código conforme a descrição acima, copie e cole na caixa de texto (a resposta da Atividade Prática sempre será em código (pseudocódigo)).

## Gabarito Atividade Prática

```
Algoritmo "Multi2"  
procedimento multi  
  var  
    aux: inteiro  
inicio  
  aux <- n * m  
  res <- aux  
fimprocedimento  
Var  
res, n, m: inteiro  
Inicio  
  Escreva("Digite um número: ")  
  Leia(n)  
  Escreva("Digite outro número: ")  
  Leia(m)  
  multi  
  escreva(res)  
Fimalgoritmo
```



**Ir para exercício**

