

Técnicas de Desenvolvimento de Algoritmos



Cruzeiro do Sul Virtual
Educação a distância

Material Teórico



Estrutura Geral de um Algoritmo

Responsável pelo Conteúdo:

Prof.^a Me. Ana Fernanda Gomes Ascencio

Revisão Textual:

Jaquelina Kutsunugi

Revisão Técnica:

Prof.^a Esp. Margarete Eliane da Silva Almendro

UNIDADE

Estrutura Geral de um Algoritmo



- Estrutura Sequencial dos Algoritmos.



OBJETIVO DE APRENDIZADO

- Desenvolver algoritmos em fluxograma e em pseudocódigo com apenas estruturas sequenciais.

Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:

Determine um horário fixo para estudar.

Mantenha o foco! Evite se distrair com as redes sociais.

Procure manter contato com seus colegas e tutores para trocar ideias! Isso amplia a aprendizagem.

Seja original! Nunca plagie trabalhos.

Aproveite as indicações de Material Complementar.

Conserve seu material e local de estudos sempre organizados.

Não se esqueça de se alimentar e de se manter hidratado.

Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como seu “momento do estudo”;
- ✓ Procure se alimentar e se hidratar quando for estudar; lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo;
- ✓ No material de cada Unidade, há leituras indicadas e, entre elas, artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados;
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e de aprendizagem.

Estrutura Sequencial dos Algoritmos

Conforme mencionado na Unidade 1, o desenvolvimento de programas é compreendido de 3 etapas: **análise** (onde estuda-se o enunciado do problema para definir os dados de entrada, o processamento e os dados de saída), **algoritmo** (onde, utilizando-se da descrição narrativa, o fluxograma ou português estruturado, descrevemos o problema com suas soluções) e, finalmente, **codificação** (onde o algoritmo é transformado em códigos de uma linguagem de programação).

Assim, nesta Unidade, vamos aprofundar-nos na etapa ALGORITMO, entendendo como cada uma de suas partes são desenvolvidas.

Podemos desenvolver ALGORITMOS apenas no papel ou fazer o uso de ferramentas que nos auxiliem neste processo e, nesse caso, usaremos a Ferramenta VisualG.



Neste link, é possível consultar uma apostila cujo autor é o desenvolvedor da ferramenta VisualG: <https://bit.ly/2SI2JHB>.

A partir deste momento, vamos explorar a estrutura sequencial de um algoritmo em fluxograma e em pseudocódigo.

Um fluxograma tem, no mínimo, 3 elementos, ou seja, o início, o processamento e o fim, seguindo os símbolos definidos na Tabela da Unidade 1.

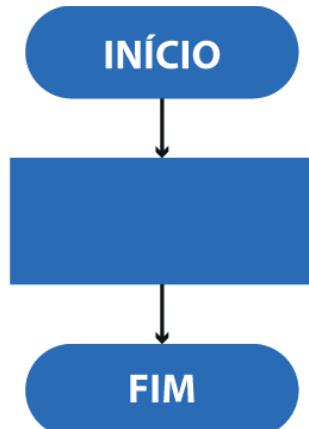


Figura 1- Estrutura sequencial em fluxograma

Já em pseudocódigo, temos uma quantidade maior de detalhes, como descrito a seguir, as seguintes partes: cabeçalho (algoritmo “nome”), comentários (sempre precedidos de //), seção de declarações de variáveis (var lista_de_variaveis: tipo), bem como as palavras início e finalgoritmo que delimitam o bloco de processamento.

```
algoritmo "nome"  
// Função:  
// Autor:  
// Data:  
// seção de declarações de variáveis e constantes  
var lista_de_variaveis: tipo  
inicio  
// processamento  
finalgoritmo
```



Neste link, é possível fazer o download da ferramenta VisualG: <https://bit.ly/46wllld>

Segundo Ascencio e Campos (2012, p. 7),

um ALGORITMO e, posteriormente, um PROGRAMA, recebe dados, tais dados precisam ser armazenados no computador para serem utilizados posteriormente no processamento. Este armazenamento é feito na memória. Desta forma, uma variável representa uma posição de memória, possuindo nome e tipo. Cujo conteúdo pode variar ao longo do tempo, durante a execução de um programa. Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante.

Os fluxogramas utilizam variáveis, mas não fazem suas declarações ou detalhamentos, já no portugol ou pseudocódigo, existe uma seção específica para suas declarações.

Esta seção deve começar com a palavra var, seguida da lista de variáveis e de seus respectivos tipos, conforme exemplo a seguir:

```
var lista_de_variaveis: tipo
```

Os tipos de dados mais utilizados são: numérico, literal e lógico, descritos a seguir, segundo Ascencio e Campos (2012, p. 8).

a) Numérico: os dados numéricos dividem-se em dois grupos: inteiros e reais.

Os números inteiros podem ser positivos ou negativos e NÃO possuem parte decimal. Este tipo de dado, quando armazenado na memória do computador, ocupa 2 bytes, por isso temos $2^8 \times 2^8 = 2^{16} = 65536$ possibilidades de representação dos números inteiros. A faixa de valores inteiros possíveis vai de -32767, -32766,, 0,, 32767, 32768.

Exemplos de dados numéricos inteiros (ASCENCIO; CAMPOS, 2012, p. 8):

-23

98

0

1350

-357

237

-2

Os números reais podem ser positivos ou negativos e possuem parte decimal. Este tipo de dado, quando armazenado na memória do computador, ocupa 4 bytes, por isso temos $2^8 \times 2^8 \times 2^8 \times 2^8 = 2^{32}$ possibilidades de representação dos números reais. A faixa de valores reais possíveis é muito maior e possui de 6 a 11 dígitos significativos com sinal.

Exemplos de dados numéricos reais (ASCENCIO; CAMPOS, 2012, p. 9):

23.45

346.89

-34.88

0.0

-247.0

OBSERVAÇÃO: “os números reais seguem a notação da língua inglesa, ou seja, a parte decimal é separada da parte inteira por um . (ponto) e não por uma , (vírgula)” (ASCENCIO; CAMPOS, 2012, p. 9).

b) Lógico: São também chamados dados booleanos (por causa da álgebra de Boole) e podem assumir os valores VERDADEIRO ou FALSO. Este tipo de dado, quando armazenado na memória do computador, ocupa 1 byte, pois possui apenas duas possibilidades de representação. (ASCENCIO; CAMPOS, 2012, p. 9).

c) **Literal ou Caractere:** São dados formados por um único caractere ou por uma cadeia de caracteres. Estes caracteres podem ser as letras maiúsculas, as letras minúsculas, os números (não podem ser usados para cálculos) e os caracteres especiais (&, #, @, ?, +). Este tipo de dado, quando armazenado na memória do computador, ocupa um byte para cada caractere. (ASCENCIO; CAMPOS, 2012, p. 9)

Exemplos de dados literais (ASCENCIO; CAMPOS, 2012, p. 9):

“aluno”

“1234”

“@internet”

“0,34”

“1 + 2”

Segundo Ascencio e Campos (2012, p. 9), os identificadores são os nomes das variáveis, dos programas, das constantes, das rotinas, das unidades, dentre outros.

As regras básicas para a formação dos identificadores são, segundo Ascencio e Campos (2012, p. 9):

- os caracteres que você pode utilizar na formação dos identificadores são: os números, as letras maiúsculas, as letras minúsculas e o caractere sublinhado;
 - o primeiro caractere deve ser sempre uma letra ou o caractere sublinhado;
 - não são permitidos espaços em branco e caracteres especiais (@, \$, +, -, %, !);
 - não podemos usar as palavras reservadas nos identificadores, ou seja, palavras que pertencem a uma linguagem de programação.
-

Exemplos de identificadores válidos, segundo Ascencio e Campos (2012, p. 10):

A

a

nota

NOTA

X5

A32

NOTA1

MATRICULA

nota_1

dia

IDADE

Exemplos de identificadores inválidos, segundo Ascencio e Campos (2012, p. 10):

5b – por começar por número

e 12 – por conter espaço em branco

x-y – por conter o caractere especial -

prova 2n – por conter espaço em branco

nota(2) – por conter o caractere especial ()

case – por ser palavra reservada

SET – por ser palavra reservada



É importante lembrar que existem regras a serem utilizadas nos identificadores e seus tipos devem possibilitar armazenagem dos dados necessários nas operações. Assim, o operador passou_na_prova: lógico está de acordo com a realidade?

Os identificadores são os nomes das variáveis e das constantes, logo, além de seguir as regras, é muito interessante colocar nomes que lembrem os conteúdos a serem armazenados, o que facilita a futura manutenção nos algoritmos e nos programas.

Segundo Ascencio e Campos (2012, p. 19), “o comando de atribuição é utilizado para atribuir valores ou operações a variáveis, sendo representado pelos símbolos a seguir”:

Quadro 1 - Comando de atribuição em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
	<-

EXEMPLO:

```
x <- 4
x <- x + 2
y <- "aula"
teste <- falso
```

Segundo Ascencio e Campos (2012, p. 19), “o comando de entrada é utilizado para receber dados digitados pelo usuário. Os dados recebidos são armazenados em variáveis”. Este comando é representado seguindo o quadro abaixo:

Quadro 2 - Comando de entrada em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
 	LEIA nome_da_variável LEIA lista_de_variáveis

EXEMPLO:

Quadro 3 - Exemplos de comando de entrada em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
X	LEIA X Um valor digitado pelo usuário será armazenado na variável X.
Y, Z	LEIA Y, Z Os valores digitados pelo usuário serão armazenados nas variáveis Y e Z.

Segundo Ascencio e Campos (2012, p. 20), “o comando de saída é utilizado para mostrar dados na tela ou na impressora”. Este comando é representado seguindo o quadro abaixo:

Quadro 4 - Comando de saída em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
	<pre>ESCREVA (nome_da_variável) ESCREVA ("mensagem", nome_da_variável)</pre>

EXEMPLO:

Quadro 5 - Exemplos de comando de entrada em fluxograma e em pseudocódigo

fluxograma	pseudocódigo
	<pre>ESCREVA (X)</pre>
	<pre>ESCREVA ("Conteúdo de Y = ",Y) Mostra a mensagem "Conteúdo de Y = " e em seguida o valor armazenado na variável Y.</pre>

Nos algoritmos, também utilizamos os operadores matemáticos e estes seguem as mesmas prioridades da matemática.

Quadro 6 - Operadores matemáticos

operador	Função
+	somar
-	subtrair
*	Multiplicar
/	Dividir

A estrutura mínima de um algoritmo consiste em receber dados, fazer o processamento e gerar uma saída. Observe o infográfico a seguir:



Figura 2 - Estrutura sequencial em algoritmos

Fonte: Levente Janos / 123RF

Desse ponto em diante, faremos alguns exemplos de algoritmos utilizando FLUXOGRAMA e PSEUDOCÓDIGO:

EXEMPLO 1 - Faça um algoritmo para calcular a média aritmética entre duas notas.

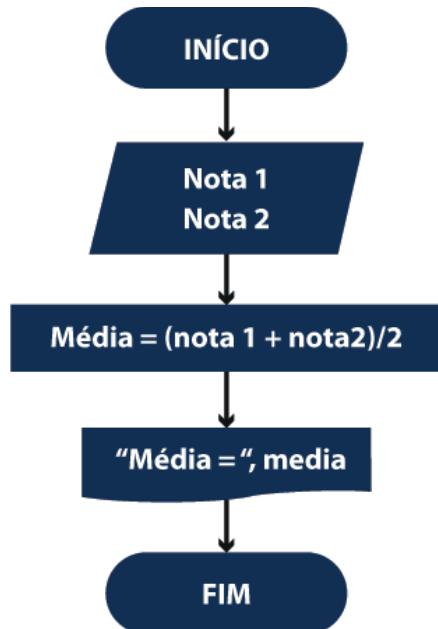


Figura 3 - Fluxograma do algoritmo para calcular a média entre duas notas

A seguir, o EXEMPLO 1 em pseudocódigo.

algoritmo “exemplo1”

```
// Função: calcular a média aritmética entre duas notas
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// seção de declarações de variáveis e constantes
var nota1, nota2, media: real
inicio
    escreval (“digite a primeira nota”)
    leia (nota1)
    escreval (“digite a segunda nota”)
    leia (nota2)
    media <- (nota1 + nota2)/2
    escreval (“Média = „,media)
fimalgoritmo
```

EXEMPLO 2 - Faça um algoritmo para calcular a média ponderada entre duas notas, cujos pesos são 2 e 3.

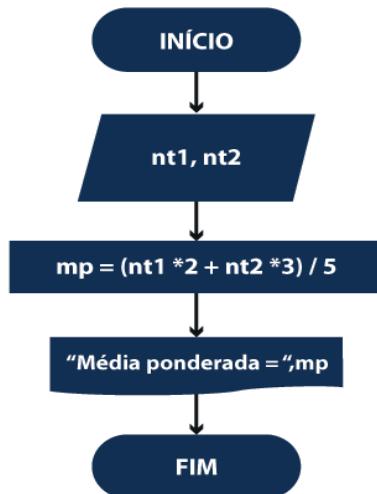


Figura 4 - Fluxograma para calcular a média ponderada entre duas notas

A seguir, o EXEMPLO 2 em pseudocódigo.

algoritmo “exemplo2”

```

// Função: calcular a média ponderada entre duas notas
// peso da primeira nota = 2
// peso da segunda nota = 3
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// seção de declarações de variáveis e constantes
var nt1, nt2, mp: real
inicio
    escreva (“digite a primeira nota”)
    leia (nt1)
    escreva (“digite a segunda nota”)
    leia (nt2)
    mp <- (nt1 * 2 + nt2 * 3) /5
    escreva (“Média ponderada = ”, mp)
fimalgoritmo
  
```

EXEMPLO 3 - Faça um algoritmo para calcular o novo preço de um produto, sabendo que o percentual de desconto será informado pelo usuário.

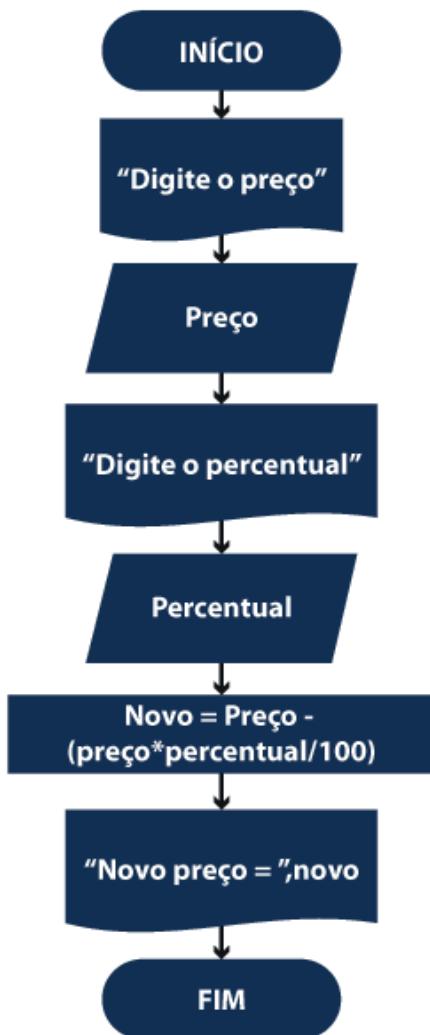


Figura 5 - Fluxograma do algoritmo para calcular o preço com desconto

A seguir, o EXEMPLO 3 em pseudocódigo.

```
algoritmo "exemplo3"
    // Função: calcular o novo preço de um produto
    // o percentual de desconto será informado pelo usuário
    // Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
    // Data: 24/12/2018      /*07/02/2020
    // seção de declarações de variáveis e constantes
    var preco, perc, novo: real
    inicio
```

```
escreva ("digite o preço ")
leia (preço)
escreva ("digite o percentual ")
leia (perc)
novo <- (preço - (preço * perc/100))
escreva ("Novo preço = ", novo)
fimalgoritmo
```

Assim, nesta Unidade, exploramos a estrutura sequencial dos algoritmos.

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:

Sites

VisualG

No site a seguir, é possível fazer download da ferramenta VISUALG, utilizada para testar algoritmos desenvolvidos em pseudocódigo.

<https://bit.ly/46wllld>

Livros

Fundamentos da Programação de Computadores

O livro “**Fundamentos da Programação de Computadores**”, no Capítulo 3, apresenta a teoria sobre Estruturas Sequenciais e vários exemplos. Ao final deste capítulo, existem duas listas de exercícios, uma resolvida e a outra a resolver.

ASCENCIO, A. F. G.; CAMPOS, E. A. V. Fundamentos da Programação de Computadores. 3. ed. São Paulo: Pearson, 2012.

Algoritmos. Teoria e Prática

O livro “**Algoritmos. Teoria e Prática**”, do autor Thomas Cormen, é um clássico na área de Computação e aborda o desenvolvimento de algoritmos na teoria e na prática.
CORMEN, T. H. Algoritmos. Teoria e Prática. Rio de Janeiro: Campus, 2012.

Algoritmos

O livro “**Algoritmos**”, de José Augusto Manzano, é muito indicado para quem está começando a desenvolver algoritmos, tendo em vista uma possível sequência didática de aprendizagem e uma linguagem apropriada para iniciantes.

MANZANO, J. A. N. G. Algoritmos. 28. ed. São Paulo: Pearson, 2016.

Referências

ASCENCIO, A. F. G.; CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**. 3. ed. São Paulo: Pearson, 2012.



Cruzeiro do Sul
Educacional