

# Arranjos

Roberto Rocha



# Programar é simples?

PUC Minas Virtual

# Você está seguindo nossas regras?

Esta estudando?



# Programar

Segundo Carlos Almeida\*, professor da Escola Secundaria Emídio Navarro, da região de Viseu, em Portugal,

“Algoritmo não é a solução de um problema, pois, se assim fosse, cada problema teria um único algoritmo”,

“Algoritmo é um caminho para a solução de um problema, em geral, os caminhos para uma solução são muitos”

O desenvolvimento de algoritmos utiliza procedimentos lógicos e de raciocínio na busca da solução de problemas.

Um mesmo problema pode ser resolvido de varias formas, tanto por uma pessoa como por um conjunto de pessoas.

Resolver um problema de varias formas não significa, em absoluto, escrever a solução de qualquer jeito.

A escrita de algoritmos deve respeitar um certo formalismo. A base do desenvolvimento de algoritmos, do ponto de vista computacional, é o próprio computador.

\*MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. **Algoritmos** : lógica para desenvolvimento de programação de computadores. 28. ed. rev. e atual. São Paulo, SP: Érica, 2016. 1 recurso online ISBN 9788536518657 – Pagina 132

# Programar

É necessário ao programador ter seu pensar modelado no computador, e para pensar "computador" é preciso seguir as regras que estamos realizando.

Programar não é trabalho simples.

Se programar um computador fosse simples, todas as pessoas o fariam e não seria necessário muito esforço para o aprendizado.

A grande dificuldade está exatamente no fato de existirem vários caminhos para a solução de um problema e ter de seguir regras de trabalho.

Como saber o caminho correto?

Qual é o melhor algoritmo para o problema?

Estude!!!

ESTUDE!!!

Pratique!!!

PRATIQUE!!!

# Programar

O professor Carlos Almeida\* adverte que o aprendizado de algoritmos não se faz copiando algoritmos, mas construindo e testando-os, ou seja, exercitando algoritmos.

Não há outra forma de fazê-lo, pois "o único lugar onde o sucesso vem antes do trabalho é no dicionário", como disse Albert Einstein.

Por mais experiência que tenha um programador, em algum momento da vida profissional terá de elaborar e testar algoritmos muitas vezes complexos e diferentes da forma que esteja acostumado.

O programador é um eterno estudante.

A construção de algoritmos voltados à programação de computadores exige cuidado e atenção.

Assim que a solução de um certo problema computacional, por meio de um algoritmo, é elaborada, é necessário realizar testes para verificar se a linha de raciocínio é correta.

\*MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. **Algoritmos** : lógica para desenvolvimento de programação de computadores. 28. ed. rev. e atual. São Paulo, SP: Érica, 2016. 1 recurso online ISBN 9788536518657 – Pagina 132

# Exercício

Faça um algoritmo para ler três notas de alunos e imprimir quantas notas tem valor abaixo da média

# Exercício

Faça um algoritmo para ler três notas e imprimir quantas notas tem valor abaixo da média

```
7 Var
8 // Seção de Declarações das variáveis
9 n1,n2,n3,media:real
10 nr:inteiro
11 Inicio
12 // Seção de Comandos
13 leia (n1,n2,n3)
14 media<-(n1+n2+n3)/3
15 nr<-0
16 se n1<media entao
17     nr<-nr+1
18 fimse
19 se n2<media entao
20     nr<-nr+1
21 fimse
22 se n3<media entao
23     nr<-nr+1
24 fimse
25 escreva (nr)
26 Fimalgoritmo
```

Imagina se fosse um programa para 60 alunos?  
Como ficaria o programa?



# Estrutura de dados homogênea

Vamos descrever uma técnica de programação que auxilia o agrupamento de dados do mesmo tipo em uma mesma variável, porém indexada.

Devido a essa característica referencia-se essa técnica como estrutura de dados homogênea (mesmo tipo de dado).

A estrutura de dados homogênea em programação recebe diversos nomes, como:

variáveis indexadas,

variáveis compostas,

variáveis subscritas,

arranjos,

vetores,

matrizes,

tabelas em memória,

arrays (do inglês) ou

conjuntos.

Para nosso estudo vamos unificar o nome de **vetor** para agrupamento de 1 dimensão e **matrizes** para agrupamentos de mais de uma dimensão.

# Matrizes de Uma Dimensão - Vetor

A matriz de uma dimensão (Vetor) é a forma mais simples de usar tabelas de valores com apenas uma coluna e varias linhas de dados.

<b>Linha</b>	<b>Coluna</b>
1	
2	
3	
4	
:	

# Matrizes de Uma Dimensão - Vetor

## Estruturas indexadas homogêneas - Vetor

Exemplo:

índice:	0	1	2	3	4	5	6	7	8	9
nome:										

onde:

nome : identificação da variável

índice: posição em que está armazenado o dado

Essa estrutura de dados fica em uma única variável dimensionada com um determinado tamanho

A dimensão de uma matriz é formada por constantes inteiras e positivas.

Os nomes dados a uma variável composta (matriz) seguem as mesmas regras dos nomes dados a variáveis simples

# Matrizes de Uma Dimensão - Vetor

## Definição de dados indexados:

A forma geral para se definir uma variável agrupada é a mesma de um variável simples, seguida pela definição de elementos do grupo, por dimensão.

Em Matlab o primeiro valor ocupará a posição de índice igual a um (1)

Em C o primeiro valor ocupará a posição de índice igual a zero(0)

Em algoritmo, tanto *<valor-inicial>* como *<valor-final>* devem ser inteiros. Além disso, exige-se evidentemente que *<valor-final>* seja maior do que *<valor-inicial>*.

## Exemplos:

vet: **vetor** [0..4] de real // reserva 5 posições de memória variável vet

qtd: **vetor** [0..9] de inteiro // vetor de 10 posições para qtd

## Em C

float vet[5]; // reserva 5 posições de memória variável vet índices de 0 a 4

int qtd[10]; // vetor de 10 posições para qtd índices de 0 a 9

# Matrizes de Uma Dimensão - Vetor

Exemplos:

v: **vetor** [0..9] de real // reserva 10 posições de memória variável V

Após preenchendo o vetor

variavel : V

indice	0	1	2	3	4	5	6	7	8	9
valor	58	4	2	123	-4	1	59	500	857	8

variavel : V

indice	0	1	2	3	4	5	6	7	8	9
elemento	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
variavel	var[0]	var[1]	var[2]	var[3]	var[4]	var[5]	var[6]	var[7]	var[8]	var[9]
valor	58	4	2	123	-4	1	59	500	857	8

# Matrizes de Uma Dimensão - Vetor

Acesso a elementos: é individual para cada elemento e deverá ser feito mediante o uso de colchetes [] envolvendo uma constante, uma variável inteira ou por expressão cujo valor resulte também em um valor inteiro.

Exemplos:

```
algoritmo "semnome"
// Função :
// Autor :
// Data :
var
  p: vetor [0..2] de inteiro
  // p vetor de 3 posicoes p[0],p[1] e p[2]
  a,b,i:inteiro
inicio
  p[0]<-5
  a<-1
  b<-1
  p[a]<-8
  p[a+b]<-13
  para i de 0 ate 2 passo 1 faca
    escreval(p[i])
  fimpara
finalgoritmo
```

P	0	5			
	1	8			
	2	13			
a		1			5
b		1			8
i		0	1	2	13

## Matrizes de Uma Dimensão - Vetor

Refazer um algoritmo para ler três notas de alunos e imprimir quantas notas tem valor abaixo da média utilizando vetores

```
7  Var
8      n1,n2,n3,media:real
9      nr:inteiro
10 Inicio
11     // Seção de Comandos
12     leia (n1,n2,n3)
13     media<-(n1+n2+n3)/3
14     nr<-0
15     se n1<media entao
16         nr<-nr+1
17     fimse
18     se n2<media entao
19         nr<-nr+1
20     fimse
21     se n3<media entao
22         nr<-nr+1
23     fimse
24     escreva (nr)
25 Fimalgoritmo
```

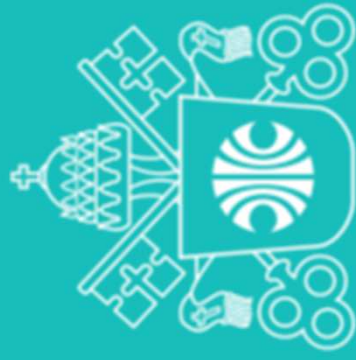
```
Var
notas: vetor [0..2] de real
media,soma:real
i,nr,tam:inteiro
Inicio
    // Seção de Comandos
    tam <-3
    soma <-0
    para i de 0 ate tam-1 passo 1 faca
        leia(notas[i])
        soma<-soma+notas[i]
    fimpara
    media <-soma/tam
    nr<-0
    para i de 0 ate tam-1 passo 1 faca
        se notas[i] < media entao
            nr<-nr+1
        fimse
    fimpara
    escreva (nr)
Fimalgoritmo
```

## Matrizes de Uma Dimensão - Vetor

### Exercícios:

Ler um vetor de 10 posições. Imprimir a soma dos valores do vetor.





# PUC Minas Virtual