



Programador de Sistemas

Lógica de Programação

Vetores – 1ª parte

Prof. Mauricio Wieler Orellana

mauricioow@gmail.com

Vetores - Introdução e definição do que é um vetor

O nosso problema começa, quando temos que declarar inúmeras variáveis para atender a um fim.

Imagine que você tem que receber o nome e a nota de 50 alunos de uma escola, e depois listar o nome de cada um e a média final de cada aluno e a média da turma.

Agora imagine você na declaração de variáveis, declarando uma a uma, as 50 variáveis para o nome, depois as 50 variáveis para as notas...

```
1  var
2  aluno1, aluno2, aluno3, ....., aluno50: caractere
3  nota1, nota2, nota3, ....., nota50: real
```

Cansativo não? Acho que se fosse realmente assim, não haveria tantos programadores quanto há hoje em dia...

É em casos como esse que entra o assunto desta aula: Vetores.

Vamos imaginar um vetor como sendo uma “caixa” com vários lugares separados para guardar seus documentos, ou seja, uma variável onde se pode armazenar mais de um dado.

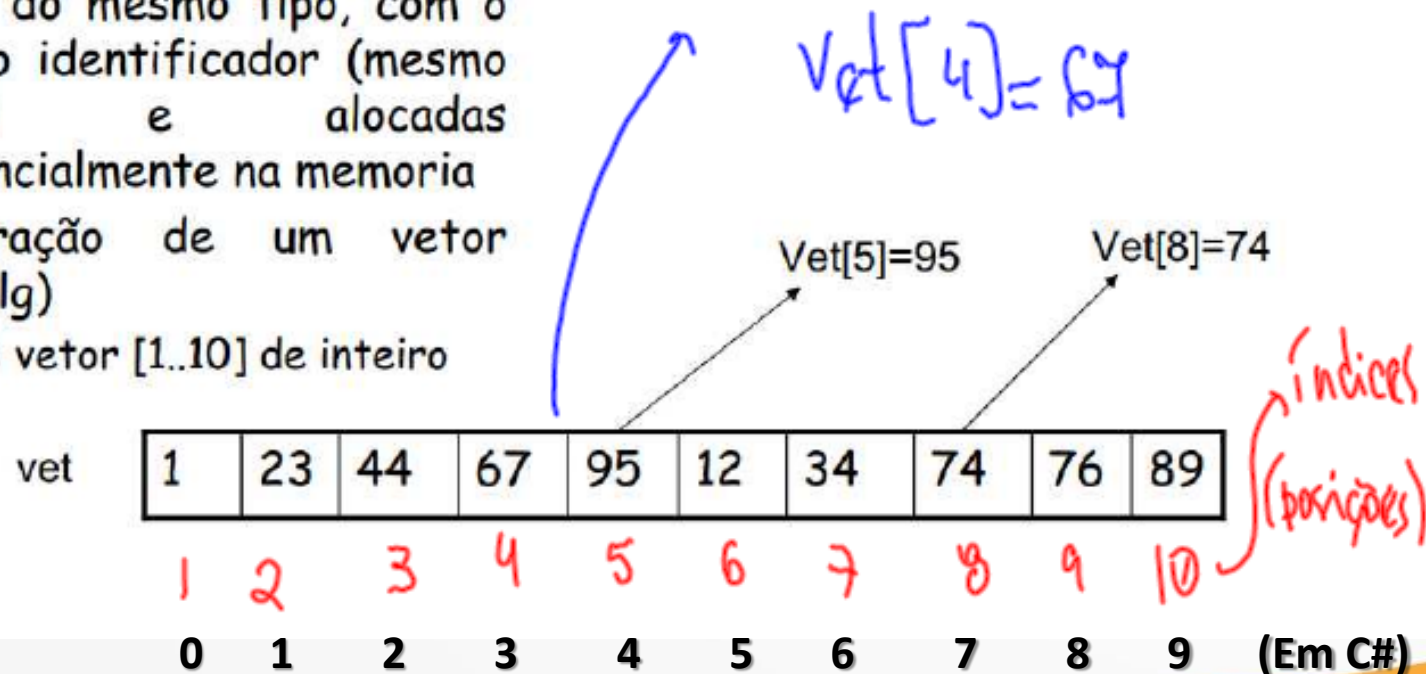


$$\begin{aligned} V[1] &= 8 & V[7] &= -4 \\ V[2] &= -1 & V[8] &= 2 \end{aligned}$$

Em C# `V[0] = 8; V[1] = -1`

Vetores

- Um vetor é uma variável composta homogênea unidimensional formada por uma seqüência de variáveis, todas do mesmo tipo, com o mesmo identificador (mesmo nome) e alocadas seqüencialmente na memória
- Declaração de um vetor (Visualg)
 - vet: vetor [1..10] de inteiro



No exemplo abaixo, temos um vetor com dados armazenados nas 5 primeiras posições:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A | B | C | D | E | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | |

meuvetor

meuvetor[4] = 'D'

Para poder acessar o conteúdo desse vetor, vamos utilizar a sintaxe: nome[posição].

Ex: `meuvetor[2] = B` `meuvetor[1] = 'B'` (Em C#)

- O retorno será a letra B.

Interessante, mas como eu posso usar vetores nos algoritmos?

Para usar os vetores, primeiro temos que declará-lo na seção de variáveis. Sim, vetor também é uma variável!

A sintaxe para declaração do vetor é a seguinte:

```
1 | var  
2 | nome-do-vetor: vetor[<posições>] de <tipo>
```

Exemplo:

```
1 | var  
2 | MeuVetor: vetor[1..5] de caractere
```

nome do vetor

Caractere
real
inteiro

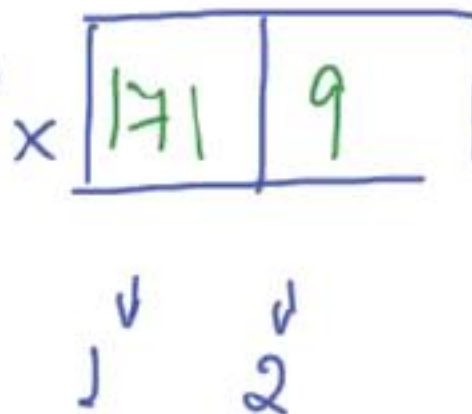
Primeiro vem o nome do vetor, então onde antes colocávamos apenas o tipo, que era caractere, real, inteiro, etc., agora vamos especificar que é um vetor, colocar quantas posições ele terá, e finalmente o tipo.

E não se confundam, no campo das posições, são apenas 2 (dois) pontos mesmo. Se colocar mais que isso seu vetor não irá funcionar. E as lendas urbanas dizem que se seu vetor estiver errado seu visualg trava. Não fiz todos os testes para confirmar isso.

$$x[1]=171 \quad x[2]=9$$

Agora, um exemplo simples para facilitar a vida de vocês:

```
1 algoritmo "vetores by algoritmizando"
2
3 var
4   x: vetor[1..2] de inteiro
5
6 inicio
7
8   escreval("Digite um número: ")
9   leia(x[1])
10  escreval("Digite mais um número =D ")
11  leia(x[2])
12  escreval("Seus números digitados foram: ",x[1]," e ",x[2])
13
14 fimalgoritmo
```



Declarando Arrays em C#

Na linguagem C# os **arrays** possuem o índice com base zero, ou seja, o primeiro elemento do array possui o índice zero (0).

Um array de uma dimensão é declarado informando o tipo de dados do array seguido do nome do array, lembrando que devemos colocar colchetes ([]) depois do tipo do array e não após o seu nome:

Ex: **int[] tabela; ==> correto** **int tabela[]; ==> incorreto**

Na linguagem C# o tamanho do array não é parte do seu tipo, isso permite declarar uma array e em seguida atribuir qualquer array de objetos int a ele, sem considerar o seu tamanho:

```
Ex:  int[] numeros;           //declara numeros como um array de inteiros de qualquer tamanho
      numeros = new int[10];   // numeros agora é um array de 10 elementos
      numeros = new int[20];   // numeros agora é um array de 20 elementos
```


1. Crie um algoritmo que leia um vetor VET de 10 posições e escreva-o na tela:

| | | | | | | | | | | |
|-----|----|----|---|---|---|----|----|----|----|----|
| VET | 12 | -1 | 8 | 0 | 4 | 23 | 12 | 15 | -1 | 13 |
| i → | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

```
algoritmo "Vetor1"
var
    VET:vetor[1..10] de inteiro
    i:inteiro
inicio
    //ROTINA DE LEITURA
    para i de 1 ate 10 faca
        escreval("Digite VET[",i,"]:")
        leia(VET[i])
    fimpara
    escreval
    escreval("O vetor digitado é o seguinte:")
    escreval
    //ROTINA DE IMPRESSÃO DO VETOR
    para i de 1 ate 10 faca
        escreva(VET[i], " ")
    fimpara
finalgoritmo
```

2. Crie um algoritmo que leia um vetor de 12 posições, de inteiros e escreva o vetor, com uma das mensagens, "é par" ou "é ímpar"

```
algoritmo "Exercício 2"
var
    VET:vetor[1..12] de inteiro
    i:inteiro
inicio
    //Rotina de Leitura
    para i de 1 ate 12 faca
        escreval("Digite o valor de VET[",i,"]:")
        leia(VET[i])
    fimpara
    escreval("Classificação de Pares e Ímpares no vetor: ")
    //Rotina de Impressão
    para i de 1 ate 12 faca
        se (VET[i]%2 = 0) entao
            escreval(VET[i]," é par ")
        senao
            escreval(VET[i]," é ímpar ")
        fimse
    fimpara
fimalgoritmo
```


3. Crie um algoritmo que leia um vetor inteiro de 10 posições, e mostre na tela :

- O vetor impresso
- Imprima os elementos pares do vetor.

```
algoritmo "vetors"  
var  
    VET:vetor[1..10] de inteiro  
    i:inteiro  
inicio  
    //ROTINA DE LEITURA  
    para i de 1 ate 10 faca  
        escreval("Digite VET[",i,"]:")  
        leia(VET[i])  
    fimpara  
    escreval  
    escreval("O vetor digitado é o seguinte:")  
    escreval  
    //ROTINA DE IMPRESSÃO DO VETOR  
    para i de 1 ate 10 faca  
        escreva(VET[i], " ")  
    fimpara  
    escreval  
    escreval  
    escreval("Os elementos pares do vetor são: ")  
    escreval  
    para i de 1 ate 10 faca  
        se (VET[i]%2 = 0) entao  
            escreval(VET[i])  
        fimse  
    fimpara  
fimalgoritmo
```

4. Crie um algoritmo que leia um vetor de salários de 8 posições e calcule e imprima o vetor com os novos salários, reajustados em 12% (aumento de 12%):

```
algoritmo "Exercicio 4"
var
    SAL:vetor[1..8] de real
    i:inteiro
inicio
    //Rotina de leitura
    para i de 1 ate 8 faca
        escreval("Digite SAL[,i,]:")
        leia(SAL[i])
    fimpara
    escreval("Salários com aumento de 12%")
    para i de 1 ate 8 faca
        //Aumento de salário em 12%
        SAL[i] <- SAL[i] + SAL[i]*0.12
        escreval("R$ ",SAL[i]:5:2)
    fimpara
finalgoritmo
```


5

● Faça um programa que carregue um vetor de seis elementos numéricos inteiros, calcule e mostre:

- ◆ a quantidade de números pares;
- ◆ quais os números pares;
- ◆ a quantidade de números ímpares;
- ◆ quais os números ímpares.

```

algoritmo "Exercicio 5"
var
    V:vetor[1..6] de inteiro
    i, contpar, contimpar:inteiro
inicio
    //Rotina de leitura
    contpar <- 0
    contimpar <- 0
    para i de 1 ate 6 faca
        escreval("Digite V[",i,"]:")
        leia(V[i])
    fimpara
    escreval
    escreval("Números Pares do vetor:")
    para i de 1 ate 6 faca
        se (V[i]%2 = 0) entao
            escreva(V[i])
            contpar <- contpar+ 1
        fimse
    fimpara
    escreval
    escreval("e existem ",contpar, " números pares no vetor !")
    escreval
    escreval("Números Ímpares do vetor:")
    para i de 1 ate 6 faca
        se (V[i]%2 = 1) entao
            escreval(V[i])
            contimpar <- contimpar+ 1
        fimse
    fimpara
    escreval("e existem ",contimpar, " números ímpares no vetor !")
finalgoritmo

```

6

● Faça um programa que carregue um vetor com dez números reais, calcule e mostre a quantidade de números negativos e a soma dos números positivos desse vetor.

2º.

1º.

3º.

Senac

V

| | | | | | | | | | |
|---|----|-----|------|------|------|---|----|-------|---|
| 3 | -5 | 8,3 | 6,36 | -5,2 | 3,23 | 4 | 20 | -4,92 | 1 |
|---|----|-----|------|------|------|---|----|-------|---|

cont_neg = 3

soma_pos = 45,89


```
algoritmo "Exercicio 6"
var
    V:vetor[1..10] de real
    somapos:real
    i, contneg:inteiro
inicio
    //Rotina de leitura
    contneg <- 0 // Contador
    somapos <- 0 //Acumulador
    para i de 1 ate 10 faca
        escreval("Digite V[",i,"]:")
        leia(V[i])
        se(V[i] > 0) entao
            somapos <- somapos + V[i]
        senao
            contneg <- contneg + 1
        fimse
    fimpara
    escreval("A soma dos números positivos vale: ", somapos)
    escreval("Existem ", contneg, " números negativos nessa coleção !")
finalgoritmo
```

7

● Faça um programa que carregue um vetor com 15 elementos inteiros e verifique a existência de elementos iguais a 30, mostrando as posições em que esses elementos apareceram.

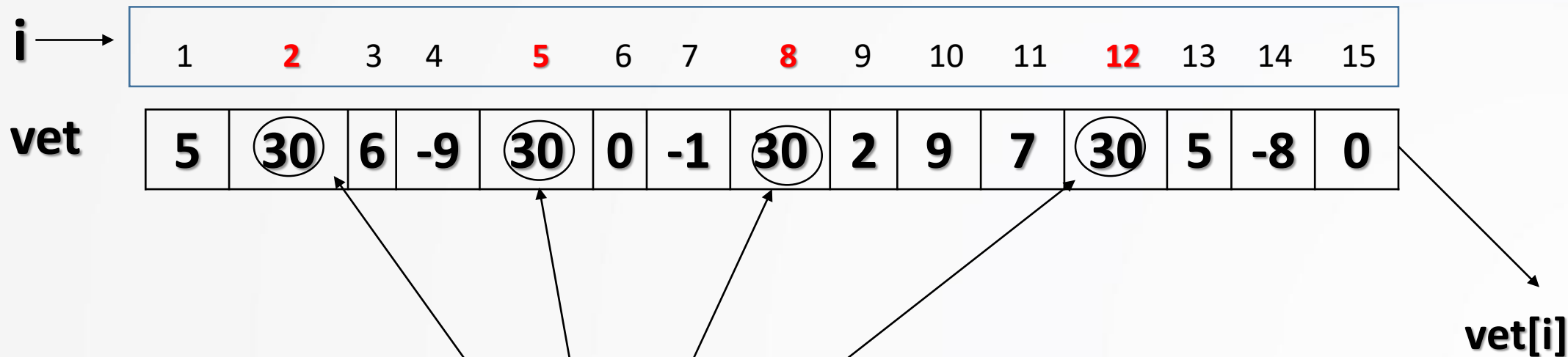
| | | | | | | | | |
|---|----|----|----|---|-------|----|----|----|
| 1 | 2 | 3 | 4 | 5 | | 13 | 14 | 15 |
| 7 | 24 | -1 | 30 | 2 | | 15 | 30 | 0 |

Existem 2 elementos iguais a 30,
na posição 4 e na posição 14

2º.

3º.

1º.



Há 4 ocorrências do elemento 30 no vetor

Nas posições:

2
5
8
12

```
algoritmo "Exercicio 7"
var
    V:vetor[1..15] de inteiro
    i, cont30:inteiro
inicio
    //Rotina de leitura
    cont30 <- 0 // Contador
    para i de 1 ate 15 faca
        escreval("Digite V[",i,"]:")
        leia(V[i])
        se (V[i] = 30) entao
            cont30 <- cont30 + 1
        fimse
    fimpara
    escreval
    se (cont30 > 0) entao
        escreval("Existem ",cont30, " números 30 no vetor, nas posições:")
        para i de 1 ate 15 faca
            se (V[i] = 30) entao
                escreval(i)
            fimse
        fimpara
    senao
        escreval("Não foram digitados números '30' no vetor!")
    fimse
finalgoritmo
```


● Faça um programa que carregue um vetor com sete números inteiros, calcule e mostre:

• os números múltiplos de 2;

• os números múltiplos de 3;

• os números múltiplos de 2 e de 3.

2 e 3 ao mesmo tempo

2º

3º

4º

1º

| | | | | | | |
|---|---|----|---|---|---|----|
| 2 | 5 | 30 | 3 | 4 | 9 | 12 |
|---|---|----|---|---|---|----|

Há 4 múltiplos de 2

Há 4 múltiplos de 3

Há 2 múltiplos de 2 e de 3, ao mesmo tempo

Senac

```

algoritmo "Exercicio 8"
var
    V:vetor[1..7] de inteiro
    i,cont2,cont3,cont2e3:inteiro
inicio
    cont2 <- 0
    cont3 <- 0
    cont2e3 <- 0
    //Rotina de leitura
    para i de 1 ate 7 faca
        escreval("Digite V[",i,"]:")
        leia(V[i])
        se (V[i]%2 = 0) entao
            cont2 <- cont2 + 1
        fimse
        se (V[i]%3 = 0) entao
            cont3 <- cont3 + 1
        fimse
        se (V[i]%2 = 0) e (V[i]%3 = 0) entao
            cont2e3 <- cont2e3 + 1
        fimse
    fimpara
    se (cont2 > 0) entao
        escreval("Os números múltiplos de 2 são: ")
        para i de 1 ate 7 faca
            se (V[i]%2 = 0) entao
                escreval(V[i]," ")
            fimse
        fimpara
    senao
        escreval("Não foram digitados múltiplos de 2")
    fimse

```

```
se (cont3 > 0) entao
    escreval("Os números múltiplos de 3 são: ")
    para i de 1 ate 7 faca
        se (V[i]%3 = 0) entao
            escreval(V[i], " ")
        fimse
    fimpara
senao
    escreval("Não foram digitados múltiplos de 3")
fimse
se (cont2e3 > 0) entao
    escreval("Os números múltiplos de 2 e de 3 são: ")
    para i de 1 ate 7 faca
        se (V[i]%2 = 0) e (V[i]%3 = 0) entao
            escreval(V[i], " ")
        fimse
    fimpara
senao
    escreval("Não foram digitados múltiplos de 2 e de 3")
fimse
fimalgoritmo
```