

Vetores

Marina Andretta

ICMC-USP

4 de abril de 2017

Um vetor (ou *array*) é uma variável composta.

Ele é formado por uma sequência de variáveis do mesmo tipo, indexadas por um inteiro que vão de 0 até o número de elementos do vetor menos 1.

Para declarar um vetor, precisamos dizer quantos elementos o vetor tem e o tipo de dados que ficará armazenado em cada posição do vetor.

Em pseudo-código, declaramos um vetor da seguinte forma:

```
<tipo> v[<tamanho>]
```

Em linguagem C:

```
<tipo> v[<tamanho>];
```

Em ambos os casos, <tipo> é o tipo de dado que será armazenado em cada posição do vetor e <tamanho> é o número de posições do vetor.

Exemplos de declarações de vetores em pseudo-código são:

inteiro v[5]

real v[2]

caracter v[10]

logico v[100]

Em linguagem C, os mesmos vetores são declarados da seguinte forma:

int v[5];

float v[2];

char v[10];

int v[100];

Quando declaramos um vetor usando

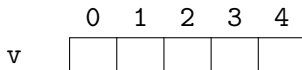
```
inteiro v[5]
```

em pseudo-código ou

```
int v[5];
```

em linguagem C, o que estamos fazendo é alocar na memória 5 espaços para variáveis inteiras, um seguido do outro.

Cada um destes espaços será acessado usando o índice correspondente do vetor.



Por exemplo, se queremos guardar o valor 7 na posição 1 do vetor v , escrevemos, em pseudo-código,

$$v[1] \leftarrow 7$$

Em linguagem C, escrevemos

```
v[1] = 7;
```

O efeito disso é

	0	1	2	3	4
v		7			

Exemplo de um algoritmo completo: em pseudo-código

```
programa media
  declaracoes
    real notas[4], soma, media
    inteiro i
inicio
  escreva(“Digite suas notas: ”)
  para i ← 0 ate 3 passo 1
    leia(notas[i])
  fimpara
  soma ← 0
  para i ← 0 ate 3 passo 1
    soma ← soma + notas[i]
  fimpara
  media ← soma/4
  escreva(“Media: ”, media, “\n”)
fim
```

Vamos simular a execução do algoritmo, supondo que o usuário digitou os números 7.3, 4.7, 10 e 6. Depois da declaração das variáveis, temos

```
soma =  
media =  
i =
```

```
      0  1  2  3  
notas 

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|


```


O primeiro laço se inicia, com o contador i começando com valor 0.

```
soma =  
media =  
i = 0
```

```
      0  1  2  3  
notas 

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|


```

Depois de lida a primeira nota (7.3), temos

```
soma =  
media =  
i = 0 1
```

	0	1	2	3
notas	7.3			

Depois de lida a segunda nota (4.7), temos

soma =

media =

i = ~~0~~ ~~1~~ 2

	0	1	2	3
notas	7.3	4.7		

Depois de lida a terceira nota (10), temos

soma =

media =

i = ~~0~~ ~~1~~ ~~2~~ 3

	0	1	2	3
notas	7.3	4.7	10	

Depois de lida a quarta nota (6), temos

soma =

media =

i = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

	0	1	2	3
notas	7.3	4.7	10	6

Em seguida, inicia-se o segundo laço, com o contador *i* iniciando com valor 0.

```
soma = 0
```

```
media =
```

```
i = 0 1 2 3 4 0
```

	0	1	2	3
notas	7.3	4.7	10	6

Teste de mesa

soma = ~~0~~ 7.3

media =

i = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~0~~ 1

	0	1	2	3
notas	7.3	4.7	10	6

Teste de mesa

soma = ~~0~~ ~~7.3~~ 12

media =

i = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~0~~ ~~1~~ 2

	0	1	2	3
notas	7.3	4.7	10	6

Teste de mesa

soma = ~~0~~ ~~7.3~~ ~~12~~ 22

media =

i = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~0~~ ~~1~~ ~~2~~ 3

	0	1	2	3
notas	7.3	4.7	10	6

Teste de mesa

soma = ~~0~~ ~~7.3~~ ~~12~~ ~~22~~ 28

media =

i = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

	0	1	2	3
notas	7.3	4.7	10	6

Teste de mesa

Por fim, encerra-se o segundo laço.

soma = 28

media = 7

i = ~~0~~ ~~1~~ 2 ~~3~~ ~~4~~ ~~0~~ ~~1~~ 2 ~~3~~ 4

	0	1	2	3
notas	7.3	4.7	10	6

E na tela aparece a mensagem:

Media: 7

Exemplo do mesmo algoritmo completo: em linguagem C

```
int main () {
    float notas[4], soma, media;
    int i;

    printf("Digite suas notas:  ");
    for (i = 0; i <= 3; i++) {
        scanf("%f", &notas[i]);
    }
    soma = 0;
    for (i = 0; i <= 3; i++) {
        soma = soma + notas[i];
    }
    media = soma/4;
    printf("Media:  %f\n", media);
    return(0);
}
```