



UFOP

**Universidade Federal de Ouro Preto - UFOP**  
**Departamento de Computação - DECOM**  
**Comissão da Disciplina Programação de Computadores I – CDPCI**  
**Programação de Computadores I – BCC701**  
**[www.decom.ufop.br/red](http://www.decom.ufop.br/red)**



# Aula Teórica 12

## Material Didático Proposto

# Conteúdos da Aula

- **Vetores**
- **Algumas Funções Aplicadas a Vetores**
- **Exercícios**

# Vetores

## Vetores

O que é um vetor na Geometria analítica?

$$k = (2, 3), m = (3, 2)$$

$k == m$ ?

Sim/Não porque?

$$v = (1, 3, 2) \text{ e } w = (-2, 4, -1)$$

Qual é o valor de  $v_1$ ,  $v_2$  e de  $v_3$ ?

E o valor de  $w_1$ ,  $w_2$  e de  $w_3$ ?

Se fizermos:  $t = v + w$ , como ficaria  $t$ ?

Ou seja, neste caso, quais são os valores de  $t_1$ ,  $t_2$  e  $t_3$ ?

Quantas dimensões (coordenadas) pode ter um vetor?

Como lidar com um vetor?

Agora vamos trabalhar com estruturas deste tipo, muito usadas na prática.

## Problema

Suponha que você deseje armazenar a nota da primeira prova de programação, de cada um dos 40 alunos de BCC701.

Seu programa teria o seguinte código para leitura de dados:

```
Nota_01 = input("DIGITE A NOTA: ");  
Nota_02 = input("DIGITE A NOTA: ");  
Nota_03 = input("DIGITE A NOTA: ");  
Nota_04 = input("DIGITE A NOTA: ");  
Nota_05 = input("DIGITE A NOTA: ");  
Nota_06 = input("DIGITE A NOTA: ");  
Nota_07 = input("DIGITE A NOTA: ");  
• • •  
Nota_40 = input("DIGITE A NOTA: ");
```

## Problema

Para a impressão de dados:

```
printf("NOTA DO ALUNO: %g", Nota_01);  
printf("NOTA DO ALUNO: %g", Nota_02);  
printf("NOTA DO ALUNO: %g", Nota_03);  
printf("NOTA DO ALUNO: %g", Nota_04);  
printf("NOTA DO ALUNO: %g", Nota_05);  
printf("NOTA DO ALUNO: %g", Nota_06);  
printf("NOTA DO ALUNO: %g", Nota_07);  
• • •  
printf("NOTA DO ALUNO: %g", Nota_40);
```

## Vetor

### Com esta codificação:

- É necessário um nome diferente de variável para cada nota de aluno, todas com o mesmo tipo de informação
- O programador deve se lembrar que a variável cujo nome é Nota\_03 representa a nota do aluno, por exemplo, João Pedro. Logo, quando o programador manipula as notas ele deve se lembrar da associação que fez entre aluno e nome de variável.
- Se o programa manipulasse 1.000 alunos, seriam necessárias 1.000 variáveis com nomes diferentes e seria “impossível” manipular todas estas variáveis.

## Uma solução:

- Seria interessante se pudéssemos armazenar todos os dados referentes às notas dos alunos em um “conjunto” com um único nome.
- Assim, todos os dados teriam um único nome de variável (o nome do conjunto) na memória.
- Para identificar cada elemento do conjunto individualmente, associaríamos ao nome do aluno um índice (por exemplo de 1 até 40).
- Você já faz este tipo de associação em Geometria. Se você tiver  $v = (v_1, v_2, v_3)$  e  $w = (w_1, w_2, w_3)$ , qual seria o resultado de  $v + w = ( \underline{\quad}, \underline{\quad}, \underline{\quad} )$ .



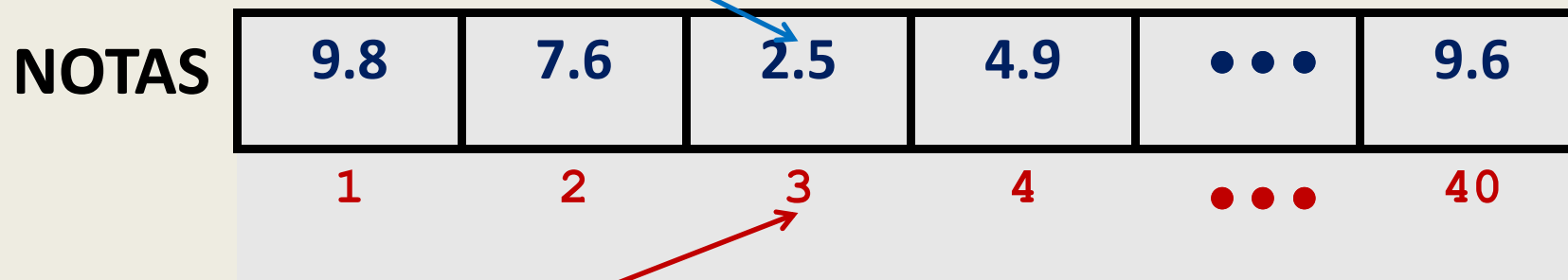
Vetor

## Uma solução:

- Em programação este conjunto de dados recebe o nome de VETOR, e no nosso exemplo teríamos um vetor com o nome NOTAS ilustrado por:

notas dos alunos (conteúdo na posição)

NOTAS	9.8	7.6	2.5	4.9	...	9.6
	1	2	3	4	...	40



índices do vetor (posição)

Assim na posição 1 de NOTAS temos o valor 9.8, na posição 2, o valor 7.6 e assim sucessivamente.

- Vetor

## Tipo de Dados Vetor

- Um vetor possui um nome, como uma variável comum, para armazenar dados na memória.
- Todos os elementos do vetor são do mesmo tipo (inteiro, real, string, booleano, etc.). Os índices são sempre inteiros, pois são as posições.
- Os elementos (conteúdos) de um vetor são unicamente identificados pelos seus respectivos índices (posições) .

Expo: NOTAS(1) contém o valor 9.8

NOTAS(2) contém o valor 7.6

NOTAS(3) contém o valor 2.5 e assim sucessivamente

## Vetor

### Tipo de Dados Vetor

- **Um vetor lembra um conjunto usado na matemática. A diferença aqui é que os elementos do vetor estão organizados em posições consecutivas e ordenadas na memória. A ordem é determinada pelos índices inteiros: 1, 2, 3, ...; não pelos dados do vetor.**
- **Um vetor representa um conjunto ordenado, pelo índice, de valores homogêneos (do mesmo tipo). Por esta razão um vetor é denominado uma Estrutura de Dados Homogênea.**

## Vetor no Scilab – Lendo as notas de exatamente 40 alunos

```
clc; clear;
for i = 1:40
    printf("\nAluno %g: ", i)      // pode ser
    nota(i) = input("nota: ");    // melhorado
end

// Para calcular a média das notas dos alunos
media = 0;
for i = 1:40
    media = media + nota(i);
end
media = media/40
printf("\n Média da sala: %g", media);
```

## Vetor Linha

- Na Entrada dos dados de um vetor, os elementos são separados por espaço em branco ou pela vírgula.

```
--> massa = [20.6 45 13.8] (atribuição)
```

```
massa =
```

```
20.6    45.    13.8    (``display``)
```

```
--> peso = [23, 44, 78] (atribuição)
```

```
peso =
```

```
23.    44.    78.    (``display``)
```

```
--> massa(1)
```

```
ans =    20.6    (``display``)
```

```
--> peso(3)
```

```
ans =    78.    (``display``)
```

## Vetor

```
--> massa = [ 20.6 45 13.8 ] (atribuição)
```

```
--> massa(1) = 0; (atribuição)
```

```
--> massa  
      0.0      45.      13.8 ('display')
```

```
--> massa(2) = massa(2) + massa(3); (operação)
```

```
--> massa (vetor modificado)
```

```
massa =      0.      58.8      13.8
```

## Vetor no Scilab – Lendo as notas de n alunos, n informado pelo usuário

```
clc; clear;
tot_alu = input("Entre com o total de alunos da turma:");
for i = 1: tot_alu
    nota(i) = input(sprintf("Alunos %g: ", i));
end

// Para calcular a média das notas dos alunos
media = 0;
for i = 1: tot_alu
    media = media + nota(i);
end
media = media/ tot_alu
printf("\n Média da sala: %g", media);
```

Outra maneira de fazer a leitura de um vetor, é quando não se sabe de antemão o total de elementos deste e quando existe uma condição de parada para a leitura dos dados.

Por exemplo, se forem digitadas as notas dos alunos de uma turma e a leitura deve ser encerrada quando for digitado um valor  $< 0$  (condição de termino da leitura e fim do vetor)



## Vetor no Scilab – Lendo as notas de um número indeterminado de alunos

```
clc; clear;
nota_digitada = input("Entre com a nota do aluno ou < 0 para
    fim: ")
cont = 0
while nota_digitada >= 0
    cont = cont + 1
    nota(cont) = nota_digitada
    nota_digitada = input("Entre com a nota do aluno ou < 0 para
        fim: ")
end
tot_alu = cont; // a var. cont contém o total de alunos
// Para calcular a média das notas dos alunos
media = 0;
for i = 1: tot_alu
    media = media + nota(i);
end
media = media/ tot_alu
printf("\n Média da sala: %g", media);
```

## Vetor no Scilab – Entrada de um vetor por lote.

```
clc; clear;
V = [110 220 330 440 550]; // entrada por
    atribuição ou em "lote", todos os valores de
    uma única vez. Não se sabe o tamanho do vetor

n = length(V); // tamanho de V, neste caso = 5
printf("\nIMPRESSÃO DOS ELEMENTOS DO VETOR\n");
for i = 1:n
    printf("V(%g) = %g\n", i, V(i));
end
printf("\n Agora a função disp(V)");
disp(V');
```

## Vetor - Definição

### Vetor Coluna

- Os elementos são separados por ponto e vírgula.

-->  $v = [11; 22; 33; 44 ]$

-->  $v =$

11.

22.

33.

44.

## Vetor

```
--> v = [11; 22; 33; 44 ]
```

```
--> v(2) = v(2) * 10;
```

```
--> v
```

```
v =
```

```
11.
```

```
220.
```

```
33.
```

```
44.
```

## Vetor - Definição

### Vetor Coluna

- Os elementos também podem ser definidos por atribuição um a um;

```
--> nota (1) = 6.8;
```

```
--> nota (2) = 7.6;
```

```
--> nota (3) = 8.8;
```

```
--> nota (4) = 9.5;
```

```
--> nota
```

```
nota =
```

```
6.8
```

```
7.6
```

```
8.8
```

```
9.5
```

## Vetor no Scilab – Verificando elementos menores que a média do vetor

```
clc; clear;
n = input("Digite o total de elementos")
media = 0
for i = 1:n
    V(i) = input(sprintf("V(%g) = \n", i));
    media = media + V(i);
end
media = media/n;
cont = 0
for i = 1:n
    if V(i) < media then
        cont = cont + 1
    end
end
printf("\n Total de valores abaixo da média %g", cont);
```

# Exercícios

# Exercícios propostos

1. Faça um programa que preencha um vetor de  $n$  elementos através de entradas do usuário **posição a posição**. Após a definição dos elementos do vetor, calcule a média dos valores e mostre o resultado.
2. Faça um programa que preencha dois vetores de 10 elementos através de entradas do usuário. Após a definição dos dois vetores, construa um terceiro vetor onde cada elemento corresponde ao dobro da soma entre os elementos correspondentes dos outros dois vetores. Imprima o conteúdo do vetor calculado.
3. Faça um programa que preencha dois vetores de 10 elementos através de entradas do usuário. Após a definição dos dois vetores, construa um terceiro vetor (20 elementos) onde os elementos de *índice ímpar* recebem os valores do *primeiro vetor* e os elementos de *índice par* recebem os valores do *segundo vetor*. Imprima o conteúdo do vetor calculado.e



# Exercícios propostos

4. Faça um programa que preencha um vetor de  $n$  elementos através de entradas do usuário **posição a posição**. Após a definição dos elementos do vetor, calcule a média dos valores e mostre o resultado.
5. Faça um programa que preencha dois vetores de 10 elementos através de entradas do usuário. Após a definição dos dois vetores, construa um terceiro vetor onde cada elemento corresponde ao dobro da soma entre os elementos correspondentes dos outros dois vetores. Imprima o conteúdo do vetor calculado.
6. Faça um programa que preencha dois vetores de 10 elementos através de entradas do usuário. Após a definição dos dois vetores, construa um terceiro vetor (20 elementos) onde os elementos de *índice ímpar* recebem os valores do *primeiro vetor* e os elementos de *índice par* recebem os valores do *segundo vetor*. Imprima o conteúdo do vetor calculado.

Entrada:  $V1 = [0.4, -1.2, 7, 12]$ ;  $V2 = [23, 98, 23.5, 45]$

Saída:  $RESP = [0.4, 23, -1.2, 98, 7, 23.5, 12, 45]$

# Exercícios propostos

7. Escreva um programa que leia um vetor **em lote**, e preencha 2 vetores, o primeiro recebe os valores pares e o segundo os valores ímpares do vetor lido.
  7. Exemplo: Ler  $V = [2, 5, 7, 85, 34, 25]$  e gerar
  8.  $V1 = [2, 34]$  e
  9.  $V2 = [5, 7, 85, 25]$
8. Escreva um programa que preencha um vetor com entradas do usuário. Considere que o usuário definirá apenas valores numéricos positivos, e que, ao desejar encerrar a definição dos elementos ele digite um valor negativo. Após a entrada de todos os elementos do vetor, calcule e imprima o seu somatório, **sem a utilização da função sum**.
9. Imprima também as posições dos valores pares do vetor lido.