

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

CONTEÚDO 04

VETORES

Disciplina: Algoritmos e POO

Professora: Alba Lopes

alba.lopes@ifrn.edu.br

<http://docente.ifrn.edu.br/albalopes>

INTRODUÇÃO

○ Variável

- Analogia: uma caixa, na qual você pode dar o nome que lhe achar conveniente, e guardar o conteúdo que desejar



- Possui um tipo (caractere, lógico, inteiro ou real)
- O valor dentro da “caixa” que pode ser alterado de acordo com a execução do algoritmo



INTRODUÇÃO

- **PROBLEMA:** Receber o nome e as 4 notas de 50 alunos de uma escola, e depois **listar** o nome de cada aluno junto com sua média.



INTRODUÇÃO

- Agora imagine como ficaria na declaração de variáveis, declarando uma a uma, as 50 variáveis para o nome, depois as variáveis para as médias de cada aluno...

```
public static void main(String [] args){  
  
    float nota1, nota2, nota3, nota4;  
    String nome_aluno1, nome_aluno2, nome_aluno3, nome_aluno4, nome_aluno5, nome_aluno6, nome_  
    float media_aluno1, media_aluno2, media_aluno3, media_aluno4, media_aluno5, media_aluno6,
```

Nem coube na tela
todas as
variáveis...

INTRODUÇÃO

- O problema começa quando se precisa declarar várias variáveis para atender a um fim.
- **PROBLEMA:** Receber o nome e as 4 notas de 50 alunos de uma escola, e depois **listar** o nome de cada aluno junto com sua média.

```
Digite o nome do aluno 1: Pedro
  Digite a nota 1 do aluno Pedro: 8
  Digite a nota 2 do aluno Pedro: 7
  Digite a nota 3 do aluno Pedro: 9
  Digite a nota 4 do aluno Pedro: 6
Digite o nome do aluno 2: Maria
  Digite a nota 1 do aluno Maria: 9
  Digite a nota 2 do aluno Maria: 8
  Digite a nota 3 do aluno Maria: 10
  Digite a nota 4 do aluno Maria: 5
Digite o nome do aluno 3:
...

```

```
**Alunos - Médias**
Pedro - 7.5
Maria - 8.0
...

```



VETORES

- Em casos como esse que é útil a utilização da **estrutura de dados** conhecida como **vetor**
- Um vetor é uma espécie de caixa com várias divisórias para armazenar coisas (dados)
 - É uma variável que pode armazenar vários valores



VETORES

meuVetor



medias



nomes



VETORES

- Os vetores são definidos pelo **tipo de dados** que eles devem armazenar e a **quantidade de posições**
- **Exemplo:**
 - Vetor de 8 posições para armazenar números reais (float)
 - Vetor de 40 posições para armazenar caracteres (String)
- Os vetores são estruturas **homogêneas**.
 - Ex: um vetor de int só armazena dados do tipo int



SINTAXE NO VISUALG

○ Declaração:

```
<tipo> [] <nome_da_variavel> = new <tipo>[<quantidade>];
```

○ Exemplo:

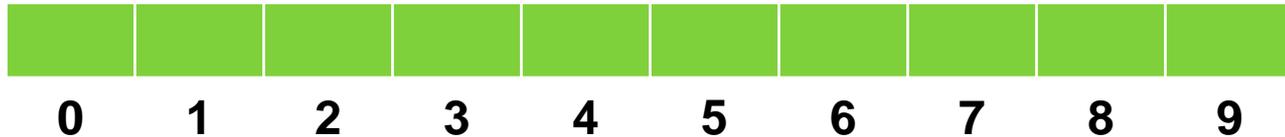
```
int [] medias_alunos = new int[50];
```



SINTAXE NO VISUALG

○ Preenchendo e acessando um vetor

- As posições dos vetores são identificadas por índices
- Em Java, as posições começam a ser contadas a partir do 0.
- Um vetor de 10 posições, por exemplo pode ser representado da seguinte forma:



SINTAXE NO VISUALG

○ Atribuição

```
<nome_variavel> [<posicao>] = <valor>
```

○ Exemplo:

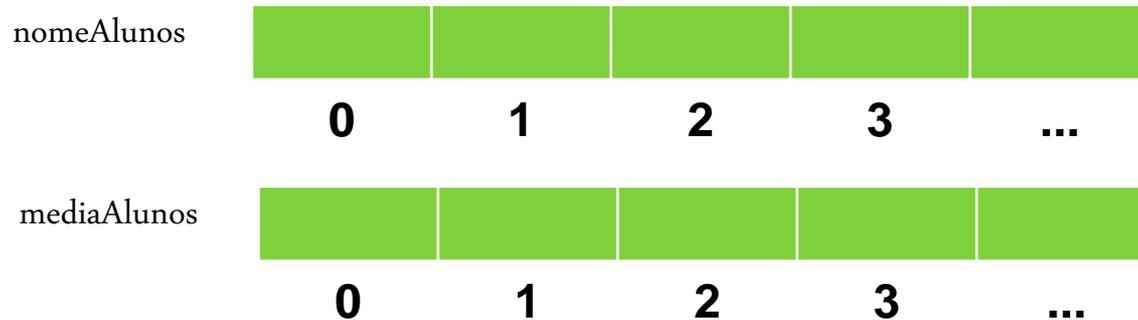
```
public static void main(String [] args) {  
    Scanner teclado = new Scanner(System.in);  
    String [] nomeAlunos = new String[50];  
    float [] mediaAlunos = new float[50];  
  
    nomeAlunos[0] = "Pedro"  
    nomeAlunos[1] = teclado.next();  
    mediaAlunos[0] = 8.3;  
    mediaAlunos[1] = teclado.nextFloat();  
}
```



SINTAXE NO VISUALG

○ Exemplo:

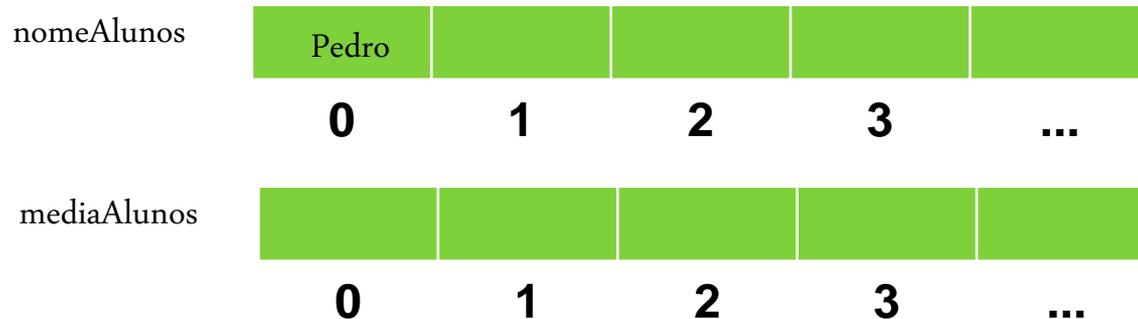
```
public static void main(String [] args){  
    Scanner teclado = new Scanner(System.in);  
    String [] nomeAlunos = new String[50];  
    float [] mediaAlunos = new float[50];  
  
    nomeAlunos[0] = "Pedro"  
    nomeAlunos[1] = teclado.next();  
    mediaAlunos[0] = 8.3;  
    mediaAlunos[1] = teclado.nextFloat();  
}
```



SINTAXE NO VISUALG

○ Exemplo:

```
public static void main(String [] args){  
    Scanner teclado = new Scanner(System.in);  
    String [] nomeAlunos = new String[50];  
    float [] mediaAlunos = new float[50];  
  
    nomeAlunos[0] = "Pedro";  
    nomeAlunos[1] = teclado.next();  
    mediaAlunos[0] = 8.3;  
    mediaAlunos[1] = teclado.nextFloat();  
}
```

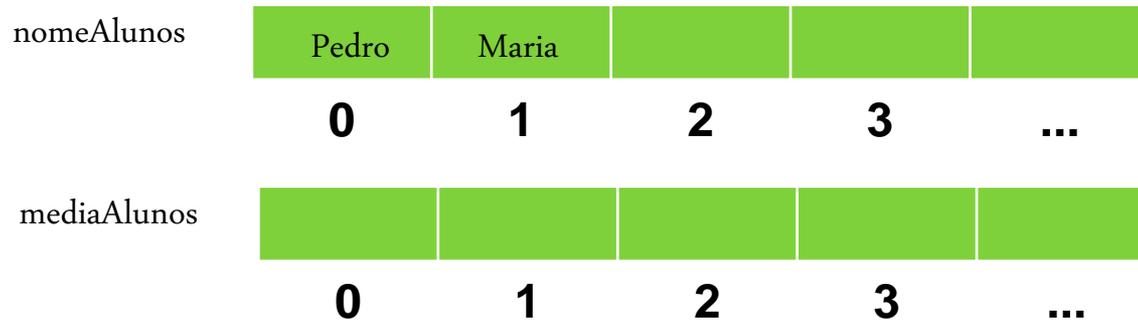


SINTAXE NO VISUALG

○ Exemplo:

```
public static void main(String [] args){
    Scanner teclado = new Scanner(System.in);
    String [] nomeAlunos = new String[50];
    float [] mediaAlunos = new float[50];

    nomeAlunos[0] = "Pedro"
    nomeAlunos[1] = teclado.next();
    mediaAlunos[0] = 8.3;
    mediaAlunos[1] = teclado.nextFloat();
}
```



SINTAXE NO VISUALG

○ Exemplo:

```
public static void main(String [] args){
    Scanner teclado = new Scanner(System.in);
    String [] nomeAlunos = new String[50];
    float [] mediaAlunos = new float[50];

    nomeAlunos[0] = "Pedro"
    nomeAlunos[1] = teclado.next();
    mediaAlunos[0] = 8.3;
    mediaAlunos[1] = teclado.nextFloat();
}
```

nomeAlunos	Pedro	Maria			
	0	1	2	3	...
mediaAlunos	8.3				
	0	1	2	3	...



SINTAXE NO VISUALG

○ Exemplo:

```
public static void main(String [] args){
    Scanner teclado = new Scanner(System.in);
    String [] nomeAlunos = new String[50];
    float [] mediaAlunos = new float[50];

    nomeAlunos[0] = "Pedro"
    nomeAlunos[1] = teclado.next();
    mediaAlunos[0] = 8.3;
    mediaAlunos[1] = teclado.nextFloat();
}
```

nomeAlunos	Pedro	Maria			
	0	1	2	3	...
mediaAlunos	8.3	7.1			
	0	1	2	3	...



SINTAXE NO VISUALG

○ Preenchendo um vetor

```
public static void main(String [] args){
    Scanner teclado = new Scanner(System.in);
    int [] numeros = new int[5];

    System.out.print("Digite um valor para a posição 0: ");
    numeros[0] = teclado.nextInt();
    System.out.print("Digite um valor para a posição 1: ");
    numeros[1] = teclado.nextInt();
    System.out.print("Digite um valor para a posição 2: ");
    numeros[2] = teclado.nextInt();
    System.out.print("Digite um valor para a posição 3: ");
    numeros[3] = teclado.nextInt();
    System.out.print("Digite um valor para a posição 4: ");
    numeros[4] = teclado.nextInt();
}
```



SINTAXE NO VISUALG

○ Preenchendo um vetor

- Podemos utilizar um laço de repetição para facilitar o preenchimento dos dados em vetores

○ Exemplo:

```
public static void main(String [] args){
    Scanner teclado = new Scanner(System.in);
    int [] numeros = new int[5];
    int i;
    for (i=0; i<5;i++){
        System.out.print("Digite um valor para a posição "+i+": ");
        numeros[i] = teclado.nextInt();
    }
}
```



SINTAXE NO VISUALG

○ Exibindo o conteúdo de um vetor:

```
...  
    System.out.println("Valor na posição 0: " + numeros[0]);  
    System.out.println("Valor na posição 1: " + numeros[1]);  
    System.out.println("Valor na posição 2: " + numeros[2]);  
    System.out.println("Valor na posição 3: " + numeros[3]);  
    System.out.println("Valor na posição 4: " + numeros[4]);
```



SINTAXE NO VISUALG

○ Exibindo o conteúdo de um vetor

- Para facilitar, podemos utilizar um laço de repetição!

○ Exemplo:

```
for (i=0; i<5;i++){  
    System.out.println("Valor na posição "+i+": " + numeros[i]);  
}
```



EXEMPLO 1

- Criar um algoritmo que leia 10 números pelo teclado e exiba os números na ordem correta em que os números foram digitados.

```
Scanner teclado = new Scanner(System.in);
int [] numeros = new int[10];
int i;
for (i=0; i<10;i++){
    System.out.print("Digite um valor para a posição "+i+": ");
    numeros[i] = teclado.nextInt();
}

System.out.println("Valores na ordem em que foram digitados correta:");
for (i=0; i<10;i++){
    System.out.print(numeros[i]+" ");
}
```

EXEMPLO 1

- Saída:

```
run:
```

```
Digite um valor para a posição 0: 20
```

```
Digite um valor para a posição 1: 4
```

```
Digite um valor para a posição 2: 6
```

```
Digite um valor para a posição 3: 15
```

```
Digite um valor para a posição 4: 8
```

```
Digite um valor para a posição 5: 99
```

```
Digite um valor para a posição 6: 7
```

```
Digite um valor para a posição 7: 23
```

```
Digite um valor para a posição 8: 8
```

```
Digite um valor para a posição 9: 12
```

```
Valores na ordem em que foram digitados correta:
```

```
20 4 6 15 8 99 7 23 8 12 CONSTRUÍDO COM SUCESSO (tempo total: 1 minuto 11 segundos)
```



EXEMPLO 2

- Criar um algoritmo que leia 10 números pelo teclado e exiba os números na ordem inversa da que os números foram digitados.

```
public static void main(String [] args){
    Scanner teclado = new Scanner(System.in);
    int [] numeros = new int[10];
    int i;
    for (i=0; i<10;i++){
        System.out.print("Digite um valor para a posição "+i+": ");
        numeros[i] = teclado.nextInt();
    }

    System.out.println("Valores na ordem inversa da que foram digitados:");
    for (i=9; i>=0;i--){
        System.out.print(numeros[i]+" ");
    }
}
```



EXEMPLO 2

- Saída:

```
run:
Digite um valor para a posição 0: 20
Digite um valor para a posição 1: 4
Digite um valor para a posição 2: 6
Digite um valor para a posição 3: 15
Digite um valor para a posição 4: 8
Digite um valor para a posição 5: 99
Digite um valor para a posição 6: 7
Digite um valor para a posição 7: 23
Digite um valor para a posição 8: 8
Digite um valor para a posição 9: 12
Valores na ordem inversa da que foram digitados:
12 8 23 7 99 8 15 6 4 20 CONSTRUÍDO COM SUCESSO (tempo total: 31 segundos)
```



EXEMPLO 3

- Escreva um algoritmo que leia um vetor com 10 posições de números inteiros. Em seguida, receba um novo valor do usuário e verifique se este valor se encontra no vetor.



EXEMPLO 3

```
public static void main(String [] args){
    Scanner teclado = new Scanner(System.in);
    int [] numeros = new int[10];
    int i, buscar;
    boolean achou = false;
    for (i=0; i<10;i++){
        System.out.print("Digite um valor para a posição "+i+": ");
        numeros[i] = teclado.nextInt();
    }
    System.out.println("Digite um número para ser buscado no vetor: ");
    buscar = teclado.nextInt();

    for (i=0; i<10;i++){
        if (numeros[i] == buscar){
            achou = true;
        }
    }
    if (achou){
        System.out.println("O número se encontra no vetor");
    }else{
        System.out.println("O número NÃO se encontra no vetor");
    }
}
```



EXERCÍCIOS

1. Crie um algoritmo que leia um vetor de 10 números inteiros. Em seguida, calcule e escreva o somatório dos valores deste vetor.
2. Escreva um algoritmo que leia um vetor com 15 posições de números inteiros. Em seguida, escreva somente os números positivos que se encontram no vetor.
3. Escreva um algoritmo que leia um vetor com 8 posições de números inteiros. Em seguida, leia um novo valor do usuário e verifique se o valor se encontra no vetor. Se estiver, informe a posição desse elemento no vetor. Caso o elemento não esteja no vetor, apresente uma mensagem informando “O número não se encontra no vetor”.



EXERCÍCIOS

5. Escreva um algoritmo que leia **dois** vetores de 10 posições e faça a soma dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Mostre o vetor resultante.

Exemplo:

vetor1	7	4	9	15	20	2	1	4	0	30
vetor2	1	8	3	7	14	9	1	8	11	16
vetorResultado	8	12	12	22	34	11	2	12	11	46

6. Crie um algoritmo que leia um vetor de 20 posições e informe:
- Quantos números pares existem no vetor
 - Quantos números ímpares existem no vetor
 - Quantos números maiores do que 50
 - Quantos números menores do que 7

